

HP AdvanceNet

Installing and Administering NFS Services

HP 9000 Computers

Installing and Administering NFS Services

**Edition 2
E0992**

**B1013-90009
Printed in U.S.A. 09/92**

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.

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

© Copyright 1992, Hewlett-Packard Company.

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 to another language without the prior written consent of Hewlett-Packard Company. The information contained in this document is subject to change without notice.

**Hewlett-Packard Co.
19420 Homestead Rd.
Cupertino, CA 95014 U.S.A.**

© Copyright 1980, 1984, 1986, AT&T, Inc.

© Copyright 1979, 1980, 1983, 1985-1990, The Regents of the University of California.

© Copyright 1979, 1986, 1987, 1988, Sun Microsystems, Inc.

This software and documentation is based in part on the Fourth Berkeley Software Distribution under license from the Regents of the University of California.

DEC and VAX are registered trademarks of Digital Equipment Corp.

UNIX is a registered trademark of UNIX system Laboratories Inc. in the U.S.A. and other countries.

Note

The Network Information Services (NIS) was formerly known as Yellow Pages (YP). The functionality of the two remains the same, only the name has changed. The name Yellow Pages is a registered trademark in the United Kingdom of British Telecommunications plc.

Printing History

New editions are complete revisions of the manual. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by the customer. The dates on the title page change only when a new edition or a new update is published. No information is incorporated into a reprinting unless it appears as a prior update; the edition does not change when an update is incorporated.

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

Edition 2 September 1992

Contents

Chapter 1 Documentation Overview

Contents of This Manual	1-2
Conventions	1-5
Documentation Guide	1-6
Military Standards and Request for Comment Documents	1-7

Chapter 2 NFS Services Overview

Components of the NFS Services	2-2
NFS Remote File Access	2-4
Named Pipes	2-6
mknod()	2-7
Device Files	2-7
NFS Mounts: Turning Off Device File Access	2-7
NFS Mounts: Mounting From NFS Device Files	2-8
Remote Execution Facility (REX)	2-9
Remote Procedure Call (RPC)	2-10
Remote Procedure Call Protocol Compiler (RPCGEN)	2-11
External Data Representation (XDR)	2-12
Network Lock Manager and Network Status Monitor	2-13
Network Information Service (NIS)	2-14
NIS Advantages	2-14
NIS Disadvantages	2-15
NIS Concepts	2-16
NIS Maps	2-17
NIS Servers and NIS Clients	2-17
NIS Domains	2-18
NIS Master and NIS Slave Servers	2-18
Virtual Home Environment (VHE)	2-20
VHE Advantages	2-20
VHE Disadvantages	2-22
How VHE Works	2-23
Example Grouping	2-24
The NFS Automounter	2-25
NFS Automounter Advantages	2-25
NFS Automounter Disadvantages	2-25
How Automount Works	2-26

Chapter 3 Installation

Key Terms	3-1
NFS Installation Checklist	3-3
Prepare the HP 9000 System	3-5
Install the NFS Software	3-6
Use update Program	3-6
Configure a New Kernel	3-8
Add a Computer to the Network	3-9

Chapter 4 NFS Configuration and Maintenance

Key Terms	4-2
Guidelines	4-5
Network Memory	4-5
Configuration Files	4-6
Daemons	4-8
Servers	4-10
NFS Configuration	4-11
Compare /etc/newconfig files to existing files	4-11
Set UIDs and GIDs	4-12
Create an NFS Server and an NFS Client Using SAM	4-14
Tips for using SAM	4-14
Move to the NFS Configuration Menu	4-14
To configure an NFS client using SAM, perform the following:	4-17
To configure an NFS Server using SAM, do the following	4-19
Create an NFS Server Manually (Without SAM)	4-20
1. Edit /etc/netnfsrc	4-20
Allow or Deny access to specific RPC services (servers) using SAM	4-23
2. Edit /etc/inetd.conf manually	4-23
3. Edit /usr/adm/inetd.sec (if necessary)	4-25
4. Edit /etc/hosts	4-27
5. Edit /etc/netgroup	4-32
6. Create and Edit /etc/exports	4-34
NFS Export Options	4-39
Directory Exports	4-41
Mount Information	4-42
7. Execute /etc/netnfsrc	4-44
Create an NFS Client Manually (Without SAM)	4-44
1. Edit /etc/netnfsrc	4-44
2. Mount File Systems	4-47
Mount Guidelines	4-48
Edit /etc/checklist for Mounts	4-53

Execute mount for Manual Mounts	4-56
3. Execute /etc/netnfsrc	4-58
Configure NIS (optional)	4-58
Configure VHE (optional)	4-58
Execute /etc/netnfsrc	4-58
NFS Maintenance	4-59
Maintain NFS Services Using SAM	4-59
To Modify or Remove Connectivity Information about a Remote System:	4-59
Prevent systems from accessing local directories or files via NFS using SAM (Stop Being an NFS Client)	4-60
Prevent remote systems from accessing local directories or files via SAM (Stop being an NFS Server)	4-61
Prevent systems from accessing local directories or files via NFS (without using SAM)	4-61
Unmount File Systems from Client	4-61
Prevent Access to Server File Systems	4-63
Update Software	4-64
Clock Skew	4-66
Maintain the NFS Server	4-69
Planned Downtime	4-69
Unplanned Downtime	4-70

Chapter 5 Remote Execution Facility (REX)

The on Command	5-2
The -i Option (Interactive Mode)	5-3
The -n Option (No Input Mode)	5-3
The -d Option (Debug Mode)	5-4
Configuration Requirements	5-4
Environment Simulation	5-5
Configuring rexd	5-6
The -l option	5-6
The -m option	5-7
The -r option	5-7
Security Considerations	5-9
Diagnostics	5-10
on Command Error Messages	5-10
rexd Error Messages	5-11

Chapter 6 The Network Lock Manager

Structure of the Network Locking Service	6-2
Starting the Network Locking Services	6-3

The Locking Protocol	6-4
The Network Status Monitor	6-5

Chapter 7 NIS Configuration and Maintenance

Key Terms	7-3
NIS Databases	7-6
Local and Global Maps	7-7
Escape Sequences	7-8
Netgroups	7-9
Files Related to NIS	7-12
NIS Commands	7-14
NIS Configuration	7-16
1. Compare /etc/newconfig Files to Existing Files	7-16
2. Create an NIS Master Server	7-17
Preparations for Creating an NIS Master Server	7-17
Restricting Access to the Master Server	7-18
Creating an NIS Master Server	7-19
Starting the NIS Master Server	7-20
3. Create an NIS Client	7-21
Creating an NIS Client	7-21
Altering a Client's Files	7-22
Starting the NIS Client	7-26
4. Create an NIS Slave Server	7-27
Preparations for Creating an NIS Slave Server	7-27
Creating an NIS Slave Server	7-27
Starting the NIS Slave Server	7-29
5. Propagate NIS Maps	7-30
6. Verify NIS	7-33
NIS Maintenance	7-34
Disable NIS	7-34
Modify NIS Maps	7-35
Manual Modifications to NIS Maps	7-36
Examples for Creating Non-Standard NIS Maps	7-37
Add or Delete a NIS Server	7-38
Add New Users to a Node	7-39
Make a Different Node the NIS Master	7-40
Create or Change NIS Password	7-41
NIS Password Installation Guidelines	7-41
NIS Password Guidelines	7-42
NIS Password	7-42
Log Files	7-43

Create Non-standard NIS Maps	7-43
Initial Example Environment	7-45
Modify ypmake	7-46
Modify Makefile	7-47
Modify ypinit	7-47
Maintain a Current Access Map on Each Slave Server	7-48
Check the Map's Contents	7-48

Chapter 8 VHE Configuration and Maintenance

Configuration Overview	8-2
1. Complete Preparation Steps	8-3
2. Compare /etc/newconfig Files to Existing Files	8-4
3. Determine File Systems and Mount Point Directories	8-4
4. Create /etc/vhe_list	8-5
Example: Simple Configuration with Single File System per Node	8-6
Example: Node with Multiple File Systems	8-6
5. Update /etc/passwd	8-7
Example: /etc/passwd file entries before and after the VHE configuration	8-7
Example: Nodes with Multiple File Systems	8-8
6. Update /etc/exports	8-8
7. Distribute /etc/vhe_list and /etc/passwd	8-9
8. Execute /usr/etc/vhe/vhe_mounter	8-9
9. Verify that VHE is Correctly Configured	8-10
Configuration Refinements	8-11
NFS mounts in the Background	8-11
VHE Maintenance	8-12
Unmounting directories or files	8-12
Adding or Deleting VHE Nodes	8-13
Advanced Usage	8-14
Adding altlogin and mounter Logins	8-14
Mounter Example	8-14
Altlogin Example	8-15
\$HOME	8-15
\$ROOT	8-15
Alternate Mount Points	8-16
Using VHE for Mail	8-16

Chapter 9 The NFS Automounter

Automount Concepts	9-2
Automount Maps	9-2

Indirect Maps	9-3
Direct Maps	9-4
Hosts Map	9-4
Automount Configuration Checklist	9-6
Planning and Design	9-7
Configure for NFS	9-7
Create Master Map	9-8
Examples	9-10
Create Direct and Indirect Maps	9-10
Indirect Maps	9-10
Special Automount Characters	9-11
Example	9-11
Direct Maps	9-12
Example	9-12
Integrating Automount With NIS	9-13
Creating NIS Maps	9-13
Command Line Options	9-16
Examples	9-17
Verify Automount Configuration	9-18
Modifying the Automount Maps	9-18
Modifying the Master Map	9-18
Modifying Indirect Maps	9-18
Modifying Direct Maps	9-18
Shutting Down Automount	9-19
Automount Error Messages	9-21
Advanced Automount Features	9-28
Replicated Servers	9-28
Hierarchical Mounts	9-28
Subdirectory Notation	9-29
Password Maps	9-30
Example	9-31

Chapter 10 Common Commands

Key Terms	10-2
NFS Commands	10-5
NFS Remote File Access	10-6
rpcinfo	10-8
rup	10-10
rusers	10-11
Example of Executing rusers	10-11
showmount	10-13

on	10-14
Network Information Service Overview	10-16
NIS Maps	10-16
NIS Servers and NIS Clients	10-17
NIS Domains	10-17
NIS Master and Slave Servers	10-17
NIS Commands	10-18
domainname	10-18
ypcat	10-19
Example of Executing ypcat	10-19
Example Using -x Option	10-20
ypmatch	10-20
yppasswd	10-21
NIS Password Guidelines	10-21
NIS Password	10-22
Example of Executing yppasswd	10-22
ypwhich	10-23

Chapter 11 Troubleshooting

Key Terms	11-2
Troubleshooting References	11-5
Power Up and Connectivity Testing	11-5
Troubleshooting Sections	11-6
Guidelines	11-7
Common Network Problems	11-7
Initial Troubleshooting	11-7
Configuration	11-8
Hardware	11-8
Network Communication	11-9
NIS and NFS Services	11-9
Remote Execution (REX)	11-10
Error Messages	11-12
Stale File Error Messages	11-12
Unsolved Problems	11-12
Flowchart Format	11-13
Troubleshooting NFS	11-14
Initial Steps to Narrowing the Problem (Flowchart 1)	11-15
Mount Fails (Flowchart 2)	11-18
Server Not Responding (Flowchart 3.1)	11-21
Server Not Responding (Flowchart 3.2)	11-24
Restricted Access (Flowchart 4)	11-28

Programs Hang (Flowchart 5)	11-31
Performance Problems (Flowchart 6)	11-34
Troubleshooting NIS	11-36
Initial Steps to Troubleshooting NIS (Flowchart 7)	11-37
Incorrect NIS Maps (Flowchart 8)	11-39
ypserv Problems (Flowchart 9)	11-42
ypbind Problems (Flowchart 10)	11-44
Multiple NIS Client Problems (Flowchart 11)	11-46
Troubleshooting VHE	11-47
Initial Steps to Troubleshooting VHE (Flowchart 12)	11-48
Home Node Goes Down After Mount Complete (Flowchart 13)	11-50
Checking /etc/passwd and /etc/vhe_list Files (Flowchart 14)	11-52
Consistency of /etc/passwd and /etc/vhe_list (Flowchart 15)	11-54
Execution of vhe_mounter (Flowchart 16)	11-56
Error Message from vhe_mounter (Flowchart 17)	11-58
Troubleshooting REX	11-59
Initial Steps to Troubleshoot REX (Flowchart 18)	11-60
Initial Steps to Troubleshoot REX (Flowchart 18.1)	11-62
Unknown Host (Flowchart 19)	11-64
Cannot Connect to REX Server (Flowchart 20)	11-66
User ID Not Valid (Flowchart 21)	11-68
User ID Denied Access (Flowchart 22)	11-70
REX Server Not Running Mount Daemon (Flowchart 23)	11-72
REX Server Denied Access through /etc/exports (Flowchart 24)	11-74
Mount Point Not a Directory (Flowchart 25)	11-77
Command Not Found (Flowchart 26)	11-79
Permission Denied (Flowchart 27)	11-81
Text File Busy (Flowchart 28)	11-83
Device files/named pipes (Flowchart 29)	11-85

Appendix A HP NFS Services vs. Local HP-UX

Appendix B Moving from RFA to NFS

Why Move to NFS Services?	B-1
Similarities	B-2
Differences	B-2
Changing Scripts from RFA to NFS	B-3
Shell Scripts that Accept Different Paths	B-3
Shell Scripts with Hard-Coded Paths	B-4
Change Directories	B-4
Create New Directories	B-5

Appendix C NFS in an HP-UX Cluster Environment

HP-UX Cluster Terms	C-1
NFS Configuration and Maintenance	C-2
Configure	C-2
Daemons	C-2
Mount/Unmount	C-2
Context Dependent Files (CDF)	C-2
Clock Skew	C-3
NIS Configuration and Maintenance	C-3
Troubleshooting	C-3

Appendix D Password Security

Glossary

Index

Documentation Overview

Before reading this manual, you should be familiar with HP-UX and have access to *HP-UX Reference* manuals.

Note The information contained in this manual applies to the HP 9000 Series 300, 400, 600, 700, and 800 computers. Any differences in the installation, configuration, operation, or troubleshooting of these computers are specifically noted.

Except for the “NIS Configuration and Maintenance” chapter, all references to servers and clients apply to NFS servers and clients unless otherwise specified.

You will find this manual helpful if you have any of the following responsibilities for the NFS (Network File System) Services product:

- Installation.
- Initial configuration of NFS, NIS (Network Information Service), VHE (Virtual Home Environment), and REX (Remote Execution Facility) services.
- Routine administration and maintenance of NFS, NIS, VHE, or REX.
- Troubleshooting common NFS, NIS, VHE, or REX problems.

Contents of This Manual

Refer to the following list for a brief description of the information contained in each chapter and appendix.

Chapter 1: Documentation Overview

This chapter describes who should use this manual, what is in this manual, and where to go for more information.

Chapter 2: NFS Services Overview

This chapter provides a brief overview of the NFS Services product, particularly the NFS, RPC, RPCGEN, REX, Network Lock Manager, NIS, and VHE services. It also describes common terms and concepts.

Chapter 3: Installation

This chapter explains how to install the NFS Services product.

Chapter 4: NFS Configuration and Maintenance

The first section explains how to set up your files in the correct configuration. It also describes NFS daemons, servers, and directories or files.

The second section explains procedures for maintaining an efficient system. It includes topics such as NFS file access removal and clock skew problems.

Chapter 5: Remote Execution Facility (REX)

This chapter explains how to configure and use the Remote Execution Facility (REX). You can use REX to execute commands on a remote host.

Chapter 6: Network Lock Manager

The Network Lock Manager and the Status Monitor permit cooperating processes to synchronize access to shared files via System V file locking primitives. This chapter describes the Lock Manager in detail.

Chapter 7: NIS Configuration and Maintenance

The first section explains how to set up your files in a configuration that allows you to centrally administer your NIS databases.

The second section explains procedures for administering and maintaining NIS. It includes topics such as modifying your system to use NIS and changing your NIS password.

Chapter 8: VHE Configuration and Maintenance

This chapter explains how to configure your system to use the Virtual Home Environment (VHE) service. VHE allows you to set up remote login environments to resemble home node login environments.

Chapter 9: The NFS Automounter

This chapter explains how to mount and unmount file hierarchies automatically using automount.

Chapter 10: Common Commands

This chapter provides brief explanations of remote file access via NFS and common NFS and NIS commands.

Chapter 11: Troubleshooting

This chapter describes how to locate and eliminate network problems, specifically those related to the NFS, NIS, VHE, and REX services.

Appendix A: HP NFS Services vs. Local HP-UX

This appendix describes the basic differences between NFS Services and local HP-UX operations.

Appendix B: Moving From RFA to NFS

This appendix describes how to translate RFA applications to NFS applications.

Appendix C: NFS in an HP-UX Cluster Environment

This appendix lists the interactions between NFS Services and HP-UX cluster nodes.

Appendix D: Password Security

This appendix explains the use of encrypted passwords and password security.

Appendix E: HP-UX 9.0 Release Changes

This appendix describes the changes made to the current release of HP's NFS services.

Glossary

The glossary lists and defines terms used in this manual that may not be familiar to you.

Index

The index provides a page reference to the subjects contained within this manual.

Conventions

Table 1-1 explains the conventions used in this manual.

Table 1-1. Conventions	
Notation	Description
Boldface	Boldface type is used when a term is defined.
Computer Text	Computer type is used for commands and keyboard entries that you must type exactly as shown. It is also used for on-screen prompts and messages.
<i>italics</i>	Italic type is used for emphasis and for titles of manuals and publications. Italic type is also used to represent a variable, such as <i>user_login_name</i> .
[Key]	This font is used to indicate a key on the computer's keyboard. When two or more keys appear together with dashes separating them, such as [Ctrl]-[D], press those keys simultaneously to execute the command.
<u>Underlining</u>	Underlining is used to emphasize a user entry. It distinguishes what you type, such as a command, from other data on the command line, such as the command prompt, a computer response, or a variable. For example: \$ date
[]	An element inside brackets in a syntax statement is optional.
...	A horizontal ellipsis in a syntax statement indicates that a previous element can be repeated. For example: <i>[option][option]...</i>

Documentation Guide

For More Information	Read
ARPA Services: Daily Use	<i>Using ARPA Services</i>
ARPA Services: System Administration	<i>Installing and Administering ARPA Services</i>
C2 Security	<i>HP-UX System Security Manual</i> <i>HP-UX Beginner's Guide</i> <i>A Beginner's Guide to Using Shells</i>
Commands and System Calls	<i>HP-UX Reference Manual</i>
Network Services: Daily Use	<i>Using Network Services</i>
Network Services: System Administration	<i>Installing and Administering NS Services</i>
Networking: General Information	<i>Networking Overview</i>
NFS Services: Common Commands	<i>Installing and Administering NFS Services</i>
NFS Services: System Administration - Configuration - Installation - Maintenance - Network Information Service (NIS) - Network Lock Manager - Remote Execution Facility (REX) - Troubleshooting - Virtual Home Environment (VHE)	<i>Installing and Administering NFS Services</i>

Military Standards and Request for Comment Documents

To obtain information about available RFCs, contact the:

Network Information Center
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025

To obtain information about available MIL-STD specifications, contact:

Department of the Navy
Naval Publications and Forms Center
5801 Tabor Avenue
Philadelphia, PA 19120-5099

NFS Services Overview

HP's NFS (Network File System) Services product allows many systems to share the same files. It is independent of the host operating system and can provide data sharing among heterogeneous systems. Explicit file transfers across the network to your local node are unnecessary. Since access techniques are transparent, remote file access remains similar to local file access.

NFS uses the client/server technology. A system running NFS can act as a client, server or both. The client makes a request to access data and files on servers. Servers make their specified resources available to the client.

- A **client** is any node or process that accesses a network service.

An NFS client can also be configured as any combination of an NFS server, NIS (Network Information Service) client, or NIS server. (An NIS server must also be configured as an NIS client.)

- A **server** is any node that provides one of the network services. A single node can provide more than one service.

An NFS server can also be configured as any combination of an NFS client, NIS client, or NIS server. (An NIS server must also be configured as an NIS client.)

Servers are passive in that they always wait for clients to call them. The degree to which clients **bind** to their server varies with each of the network services. However, the client always initiates the binding. The server completes the binding subject to access control rules specific to each service.

NFS servers are **stateless**; they do not maintain information relating to each client being served. Each file request goes to the appropriate server with the parameters attached to it locally (e.g., read and write privileges). One advantage of servers being stateless is that it is possible to reboot servers without adverse consequences to the client.

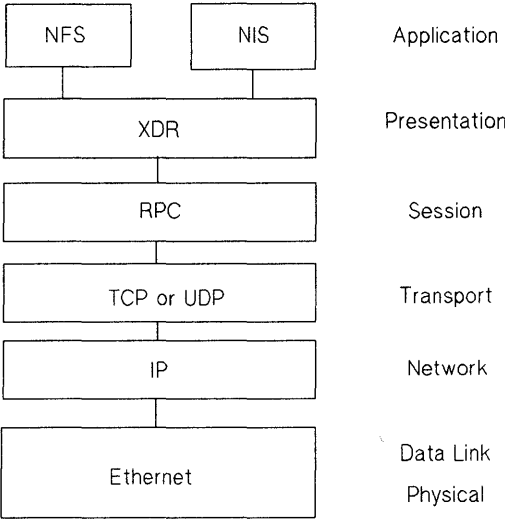
Components of the NFS Services

The NFS Services product includes the following components:

- NFS remote file access.
- Remote Execution Facility (REX).
- Remote Procedure Calls (RPC).
- Remote Procedure Call protocol compiler (RPCGEN).
- External Data Representation (XDR).
- Network Lock Manager.
- Network Status Monitor.
- Network Information Service (NIS).
- Virtual Home Environment (VHE).
- The NFS Automounter.

The NFS, REX, Lock Manager, and NIS functionalities are built on top of RPC and XDR library routines.

The NFS protocol stack is not an exact fit to the specifications of the ISO services model, but there is a close similarity between the two which provide a good framework for visualizing how the various protocols fit together.



The NFS Services Protocol Stack

NFS Remote File Access

Before the client can access remote files, the following steps must be done:

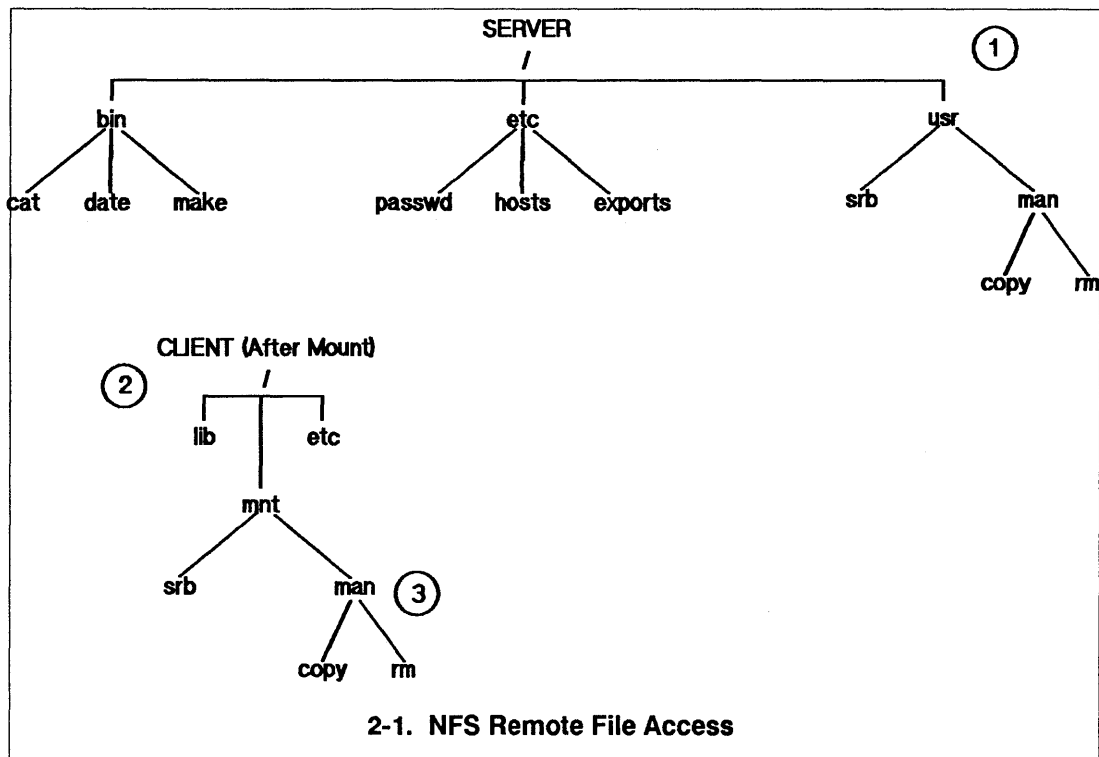
- On the server, the superuser must export the directory (i.e., make it available) to the client.
- On the client, the superuser must mount (import) the directory.

Note Like local HP-UX operations, if you copy files from a long file name directory to a short file name directory, then file names longer than 14 characters will be truncated after the 14th character.

Long and short file name directories or files are set up by the System Administrator.

Access to remote files is the same as for local files. You need to include either the complete path name starting with / (slash) or the path name relative to the current directory. The following figure and steps explain how NFS remote file access works.

EXAMPLE:



1. The superuser edits the server's `/etc/exports` file to export the `/usr` directory.

```
server superuser% cat /etc/exports  
/usr access = client_name
```

2. The superuser then runs the `exportfs` utility to make the `/usr` directory available to the client.

```
server superuser% exportfs -a
```

3. On the client, the superuser creates a mount point `/mnt` (empty directory) and mounts the directory.

```
client superuser% mkdir /mnt  
client superuser% mount server:/usr /mnt
```


4. The client reads the files in the /mnt directory.

```
client% more /mnt/man/copy
```

Two very important features of NFS Remote File Access are **named pipes** and **device files**. The following sections explain the details of these two features.

Named Pipes

A named pipe is a special type of object in the HP-UX directory. A named pipe is one of the many ways in HP-UX that unrelated processes can communicate. HP-UX processes executing on the same client system are able to communicate using named pipes. You can use named pipes via normal file operations, e.g. `open()`, `close()`, `read()`, `write()`. Typically, one process will open the named pipe for reading and another process will open it for writing.

To illustrate named pipes, consider the following example:

EXAMPLE:

C1 and C2 are processes executing on system C. Also assume host C has mounted directory / from host S on /mnt. C1 opens /mnt/FIFO for reading and C2 opens /mnt/FIFO for writing. C1 can now read what C2 wrote to the named pipe.

Next, assume a third process (process D3) is running on another client D which also has / from S mounted on /mnt (on system D), and it opened /mnt/FIFO for reading. Is process D3 able to read what process C2 wrote to this named pipe? No, because no actual NFS activity occurs between the NFS client and NFS server for named pipe reads and writes. These are handled entirely by the client.

Note In certain cases there would be NFS activity. For example, if you do a `chown` on the named pipe, the request will go to the server to change the owner.

mknod()

Named pipes are created with `mknod()`. Any user can create a named pipe with `mknod()`. (Use of `mknod()` to create device files requires superuser privileges.)

Note If you attempt to make a directory or a network special file over NFS, `mknod()` will fail and will return with `errno` set to `EINVAL` (invalid argument).

Device Files

Device files are another type of object in the directory, and are used to access physical or conceptual devices attached to the system. NFS device files always refer to a device attached to the local system and can generally be used where a local device file would be used. Like named pipes, device files are operated on through normal directory operations. For example, to write to the system console, you can write to the file `/dev/console`.

To illustrate the use of device files, consider the following:

EXAMPLE:

System C is an NFS client of the NFS server system S, and has mounted directory / from host S on `/mnt` (a superuser on system C executed the command `mount S: /mnt`). If a process on system C attempts to write to `/mnt/dev/console`, a device file representing the system console on system S, the output will go to the system console on system C, not on system S. If a process on system S attempts to write to `/dev/console`, which is the same “file” that system C wrote to, it will actually write to the console on system S.

NFS Mounts: Turning Off Device File Access

NFS device files are not secure. Therefore, the system administrator has the option of turning off device file access on a per-NFS mount basis. The administrator uses the `-o nodevs` option to the mount command to turn off device file access.

EXAMPLE:

```
mount -o nodevs nfsserver:/servermountpoint /clientmountpoint
```

Note The `nodevs` option does not turn off support of named pipes.

NFS Mounts: Mounting From NFS Device Files

You may mount a local disk that is represented by a remote NFS device file.

EXAMPLE:

```
mount /mnt/nfs/dev/dsk/0s0 //localmntpt
```

Access to the newly mounted directory will proceed as if the disk had been mounted from a local device file.

Note Access to the local disk's mounted directory will not be affected even if the NFS directory is unmounted.

Normally when unmounting a directory, you can give either the name of the device file or the name of the mount point. However, if the NFS server is down or the NFS directory is down, you must give the mount point to unmount the local disk.

EXAMPLE: You would enter the following to unmount a local disk:

```
umount //localmntpt
```

instead of:

```
umount /mnt/nfs/dev/dsk/0s0
```

The latter case will not fail if the NFS server is down, but it will hang until the server comes back up as any other NFS access does.

Remote Execution Facility (REX)

The Remote Execution Facility allows you to execute user commands and programs over the network. It provides access to compute capabilities not available on the local system by extending the user's ability to run commands and programs remotely. REX is similar to the Berkeley Service remote shell (`remsh`) with two major differences:

- Your environment is simulated on the remote host.
- You can execute interactive commands on the remote host.

Remote Procedure Call (RPC)

NFS Services runs on top of the RPC (remote procedure call). It uses the RPC at the session layer of the network. The RPC mechanism provides the ability for clients to transparently execute procedures on remote systems of the network. Usually, the system on which the procedure call is executed has resources that are not available on the requesting system. The system owning the resource becomes a server and the requesting system becomes a client of that server when it needs to access that resource.

NFS clients access server information and processes by making a remote procedure call. RPC allows a client process to execute functions on a server via a server process. Though these processes can reside on different network nodes, the client process does not need to know about the networking implementations.

To initiate an RPC, the client creates a session by locating the appropriate server. The client system then sends an encoded message to the server. This message includes all the data needed to identify the service and user authentication information. If the message is valid (i.e., calls an existing service and the authentication passes) the server performs the requested service and sends a result message back to the client.

Remote Procedure Call Protocol Compiler (RPCGEN)

RPCGEN is a Remote Procedure Call compiler. You use it to convert applications running on a single computer to ones that run over a network. It is also used to assist in writing Remote Procedure Call applications simply and directly. With RPCGEN, your development time will be reduced and you will spend less time coding and debugging network interface code.

To convert an application to run on a network, you need to produce three files. These files are:

- Protocol description file.
- Client side file.
- Server side function file.

RPCGEN accepts remote program interface definitions (the protocol description file) written in RPC and produces the following C output files, which you may use as a starting point, rewriting as necessary:

- Header file.
- Client side subroutine file.
- Server side skeleton file.
- XDR (External Data Representation) routine file.

External Data Representation (XDR)

A common data representation is required if applications are to run transparently on heterogeneous networks or if data is to be shared among heterogeneous systems. XDR is the universal data representation used by all nodes. Each node translates machine dependent data formats to XDR format when sending and translating data. Rather than execute a procedure at the local system, RPC bundles the procedure from machine dependent data formats (i.e., internal representations) to a universal format used by all network nodes using the RPC/XDR. The actual bundling is handled at the presentation layer, by the eXternal Data Representation (XDR) functionality. It is XDR that enables heterogeneous nodes and operating systems to communicate with each other over the network.

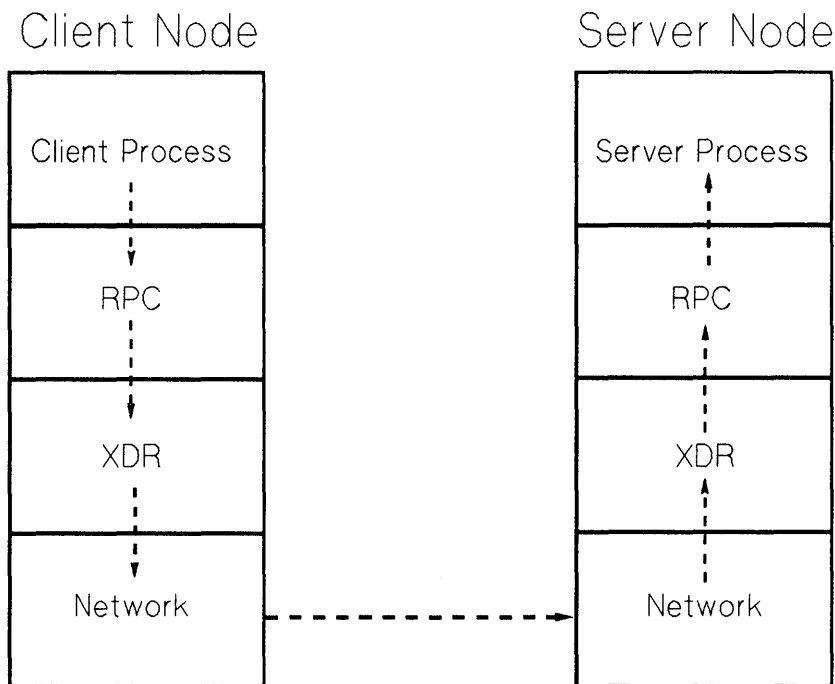


Figure 2-2. RPC and XDR Data Transfer

Note This figure does not correspond to the ISO Model.

Network Lock Manager and Network Status Monitor

NFS Services includes the Network Lock Manager (`rpc.lockd`) and the Network Status Monitor (`rpc.statd`). The Network Lock Manager supports file locking and synchronized access to shared files via `lockf` and `fcntl` for NFS. It allows users to coordinate and control access to information on the network. The Network Status Monitor is used by the Network Lock Manager to maintain the stateful locking service within the stateless NFS environment. The Network Status Monitor can also be used by applications to monitor the status of other computers and systems.

Network Information Service (NIS)

The Network Information Service (NIS), previously called Yellow Pages (YP), a registered trademark of British Telecommunications, is a distributed database system that allows commonly used configuration files to be propagated throughout the network using a centralized management facility, the NIS master server. NIS includes the following features:

- Easier system administration of network configuration files. Typical configuration files include: `/etc/group`, `/etc/hosts`, `/etc/netgroup`, `/etc/networks`, `/etc/passwd`, `/etc/protocols`, `/etc/rpc`, and `/etc/services`.

For example, programs previously read `/etc/hosts` to find an Internet address that corresponds to a host name. Previously, when you added a new node to the network, you had to add a new entry to every node's `/etc/hosts` file. Now you update the corresponding master NIS map and programs can use NIS to obtain information from NIS servers. One database is maintained for each file on the NIS master server.

- Information consistency. Since the NIS master server propagates all databases (maps) to the **slave servers**, an NIS client receives consistent information regardless of which NIS server it accesses.
- Availability. If a remote node running an NIS server process crashes, NIS client processes can obtain NIS services from another NIS server.
- Interoperability with other vendor's systems because the NIS interface uses RPC and XDR, the service is available to other vendors.

NIS Advantages

NIS has several advantages:

- NIS enables you to automatically keep user IDs and group IDs consistent among all the nodes participating in NFS file sharing.

Without NIS, you have to manually keep these IDs consistent for NFS.

- NIS provides the convenience of centrally administering the `/etc` files: `group`, `hosts`, `netgroup`, `networks`, `passwd`, `protocols`, `rpc`, and `services`.

Without NIS, you must administer these files individually on each node.

- In combination with the NFS automounter, a consistent network-wide view of the directory can be achieved for all clients.

NIS Disadvantages

NIS has the following disadvantages:

- If a network grows beyond 2000 nodes, NIS may begin to exhibit poor performance or failures. (This limit is based on today's system capacity.)
- Because NIS provides NIS clients access to data via the network, NIS clients may observe slower performance than if the data were accessed from local files. For example, when using NIS for password administration, logging in may take more time if the NIS server is busy.
- If any of the NIS servers are unstable, remote access to files may be slower since the NIS clients may have to rebind to another NIS server. If no other NIS server is available, users may not be able to login to their nodes without access to the NIS's passwd map.
- NIS does not make changes visible to all users unless the changes are made on the NIS master server.
- The NIS slave servers do not immediately see the changes made to the NIS master server maps. The updated maps become consistent among all NIS servers only after each slave server successfully copies the maps via `ypxfr`.

Note If you configure the BIND Name Server, it will be used instead of NIS for host name and address resolution. NIS will still be used for all other information such as passwords. See "Configuring and Maintaining the BIND Name Server" in the *Installing and Administering ARPA Services* manual.

NIS Concepts

Refer to the following figure and subsections for a summary of how components within the Network Information Service work together: maps, NIS domains, NIS servers (masters and slaves), and NIS clients.

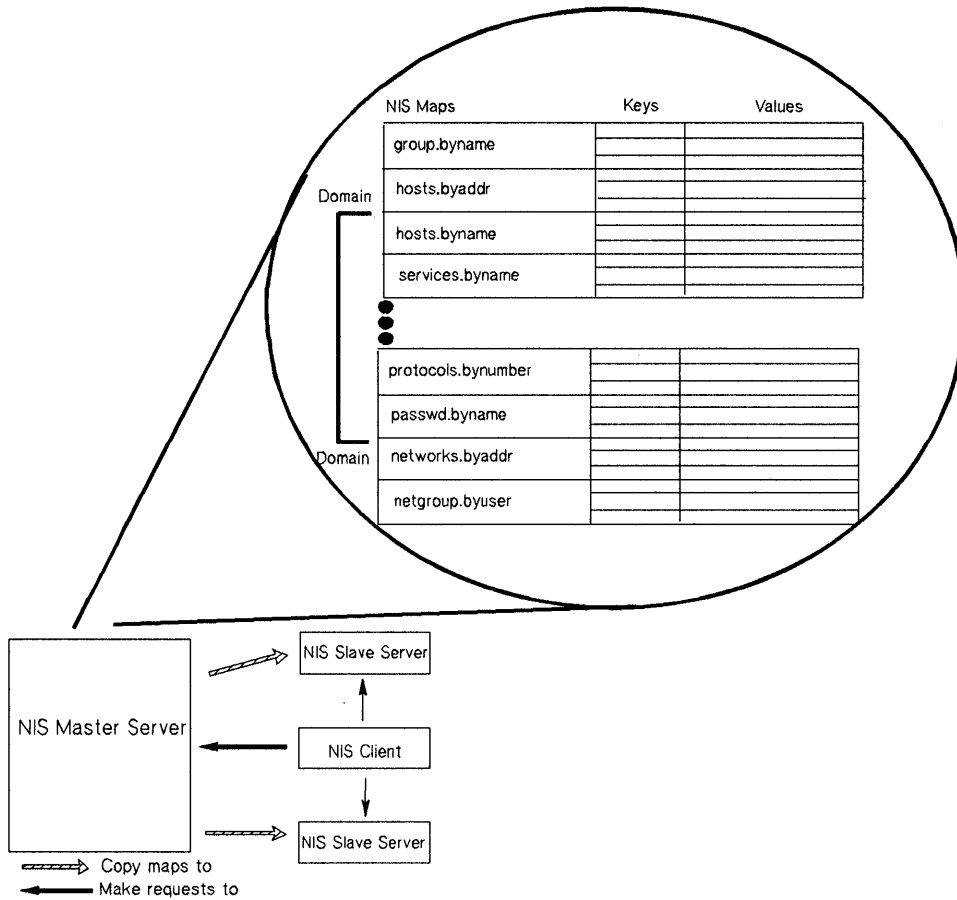


Figure 2-3. Network Information Service Structure

NIS Maps

The NIS system stores information in NIS maps (databases). Each map contains a set of keys and associated values: one key per value and one value per key. (A value may be a string of characters with imbedded blanks or tabs). For example, in the `passwd.byname` map, all the login names are the keys and their matching lines from `/etc/passwd` are the values.

Each map has a unique **map name** that programs use to access the map. Programs must know the format of the data in the map. Many of the maps are derived from ASCII files such as `/etc/hosts`, `/etc/group`, and `/etc/passwd`. The map format is usually identical to the ASCII file format.

NIS Servers and NIS Clients

NIS servers are nodes that provide access to NIS maps via the network. These maps are in `/usr/etc/yp` subdirectories named after the appropriate NIS domains. (See the next section, “NIS Domains.”)

NIS clients are nodes that request access to NIS maps from an NIS server as follows:

1. An NIS client that is not bound sends a broadcast to all NIS servers on the network.
2. The NIS client binds to the first NIS server that responds. (Each NIS client binds to one NIS server per NIS domain.)
3. If the request is the NIS client's first attempt to access data, the NIS client remembers which NIS server responded to the request. Subsequent requests by this NIS client go directly to this NIS server.
4. If the bound NIS server is down or unavailable, the NIS client automatically rebinds to the first NIS server that responds to another broadcast.

Note An NIS client can also be configured as any combination of an NIS server, NFS client, or NFS server.

An NIS server must be configured as an NIS client. It can also be configured as an NFS server, NFS client, or both.

NIS Domains

An **NIS domain** is a logical grouping of the set of maps contained on NIS servers. The following rules apply to NIS domains:

- Nodes that belong to the same NIS domain have the same domain name.
- An NIS domain has only one master server.
- An NIS domain may have zero or more slave servers.
- Maps with the same name in different NIS domains can have different contents.

You implement an NIS domain as a subdirectory of `/usr/etc/yp` on each NIS server; the name of this subdirectory is the name of the NIS domain. For example, maps in the **research** NIS domain would be in `/usr/etc/yp/research`. (Note: NIS domain names are case sensitive.) All directories that appear under `/usr/etc/yp` are assumed to be domains that an NIS server serves. To remove a domain being served, you must delete that domain's subdirectory name from `/usr/etc/yp` on all of its servers.

The `/etc/netnfsrc` file usually contains the default NIS domain name. You can change the default by executing the `domainname` command or by editing the `/etc/netnfsrc` file and then rebooting the system.

NIS Master and NIS Slave Servers

Only two types of nodes have NIS databases: master and slave servers.

The **NIS master server** is the node on which NIS maps are built from ASCII files. It is this node that contains the master databases (maps) which other NIS servers (slaves) copy. Note that the NIS master server may also provide NIS clients access to NIS maps.

Note

You should create and modify NIS databases only on the NIS master server; otherwise, all NIS databases will not be consistent across the NIS servers.

The **NIS slave servers** are the nodes that receive the propagated maps from the NIS master server. In turn, they provide NIS clients access to NIS maps.

An NIS server can be the master or slave of many domains. However, an NIS server can only be either the master or a slave of a given domain.

Though an NIS server may be master for one map and slave for another, random assignment of maps to NIS master servers may cause confusion. Therefore, only one NIS server should be the master for all maps within an NIS domain.

Virtual Home Environment (VHE)

Virtual Home Environment (VHE) is an HP-developed service that allows you to configure your login environment on remote nodes to mirror the login environment on your home node. (Home node refers to the node on which your home directory physically resides.) VHE is an optional service that is available to any HP-UX system that has the NFS product. It may also be used with other UNIX systems that support symbolic links and NFS.

If you find that you never need to work from a remote node, you may want to skip this section.

VHE Advantages

VHE's major advantage is that you can sit down at any remote node (assuming you have login permission), login, and enter into the work environment that is associated with the login on your home node (your home directory as specified in `/etc/passwd`). This includes:

- Home shell configuration (i.e., whichever shell you are configured to use on your home node appears when you login to a remote node).
- Access to files on the directories or files exported for VHE on any computers connected with VHE on the network to which you have a login and file access permission.
- Use of previously defined aliases (only for C or K shells) and shell variables.
- Use of customized shell scripts (assuming shells operate similarly on your home node and the node you are currently using).
- Use of compiled files under your home directory from your home node (assuming your home node and the node you are logged into are of the same architecture and operating system).

Thus, VHE allows you to minimize the number of computer interfaces you must learn to be productive on the various computers that are running NFS on your network and you are no longer tied to a particular computer to complete your work tasks.

Another advantage of VHE is that it distributes computational work more efficiently between nodes than ARPA/Berkeley terminal emulation services such as telnet or rlogin. Unlike telnet or rlogin, VHE does not return to your home node, that contains your home environment login, to execute tasks. Instead, VHE takes advantage of the computing capacity of the machine you are currently using. For example, if you use VHE on a node other than the home node and perform an ls command of a directory on the home node, the ls command is executed from the local /bin directory. VHE does not return to your home node's /bin directory to execute the ls command. The following figure illustrates this concept.

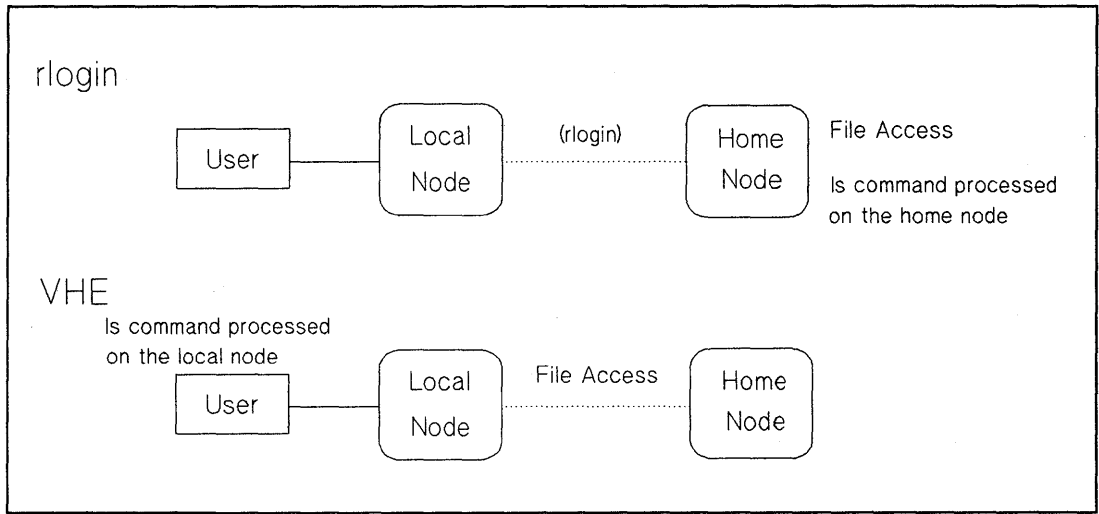


Figure 2-4. VHE vs. rlogin Performing ls Command

VHE Disadvantages

VHE has the following disadvantages:

- Though you can edit source code files originating from different types of computers on the network, you will not be able to execute object code files from a computer of a different architecture using VHE. For example, consider the following: You are currently working on an HP 9000 Series 300 and running VHE, and your home node is an HP 9000 Series 800 computer. If you try to execute an object code file on the HP 9000 Series 300 from the Series 800 computer it will not succeed. However, you can execute a script from the Series 800 computer.
- If you specify directories or hardware attributes in your node's `.profile` or `.login` files, you may have to modify these files to use VHE effectively. For example, the `.login` file needs to prompt for the terminal type if you plan to use VHE from more than one terminal or display type. If you do not already have this capability, then look in the sample `/etc/d.login` or `/etc/d.profile` files for samples of how to do this.
- When you are in your home environment, you may execute set-uid root programs that access files in your home directory. These files must allow access for the user "nobody." If this is not done, set-uid root programs will fail. The same applies for root access via set-uid. For example, your home directory is accessed via VHE and you execute set-uid to gain superuser privileges. If your shell happens to be ksh, your root ksh may hang if your `.history` file does not allow access for user "nobody."

How VHE Works

The following diagram illustrates the directory structure of nodes in a network using VHE.

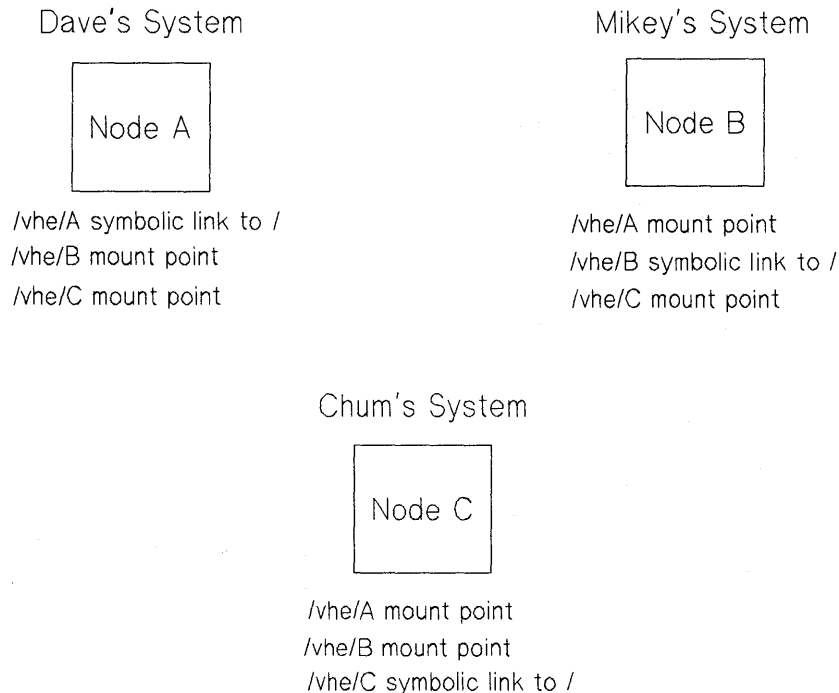


Figure 2-5. Directory Structures of Nodes Using VHE

Each node is connected to the other nodes via NFS Services. In the picture, each node is a home node for a different user (Dave, Mikey and Chum). Each user has a customized work environment set up by the login process. Directories on each home node correspond to each of the remote nodes. For example, on node A there is a directory `/vhe/B` that corresponds to node B. Using these directories as mount points, a mount is done by each node to each remote node. (The definitions of mounts and mount points are included in the "Glossary." More detailed information is contained in the "NFS Configuration and Maintenance" chapter in this manual).

Using VHE gives each node access to directories or files located on the remote nodes. To maintain consistency when you log into your home node, a symbolic link (a pointer) points to the host's root directory.

In a single node HP-UX configuration, the `/etc/passwd` file contains the directory that becomes the home directory for the user upon logging in. For use with VHE, `/etc/passwd` is edited such that all of the home directories are prefixed with a mount point or a symbolic link.

When the login program performs a `cd` to the user's home directory, the `cd` and subsequent requests are made to the user's home node via NFS Services unless logging in on your home node.

Example Grouping

In the `/etc/passwd` file, the appropriate mount point or symbolic link is added to the beginning of the directory of the home directory for each user. The example below shows how the lines in `/etc/passwd` would look for the users Dave, Mikey, and Chum as shown in Figure 2-5:

```
dave::117:100:Dave:/vhe/A/users/dave:/bin/csh
mikey::118:100:mikey Pom :/vhe/B/users/mikey:/bin/sh
chum::119:200:chum Pom:/vhe/C/users/chum:/bin/ksh
```

No matter which node Dave logs in on, his home directory is `/users/dave` on node A. When scripts such as `.login` or `.cshrc` are executed, they define the execution environment as customized by Dave. His files, shell variables, and aliases are available just as if he had physically logged in on node A.

Because VHE is not a virtual terminal program, when Dave executes processes, they are executed on the node he is logged into. If he is on node B, processes are executed on node B, not his native host A. For example, consider the following. Dave is working at node B and his system administrator has configured VHE to be running. Dave enters the following command on node B:

```
cc testfile.c
```

The `cc` from node B's `/bin` directory is executed, but `testfile.c` is used from Dave's current working directory on node A.

The NFS Automounter

Automount is an NFS tool which allows users to mount and unmount remote files and directories on an as needed basis. Mount information including the server's name, directory on the server, local mount point and mount options, are contained in automount maps. If a certain amount of time passes and the directory has not been used, it is automatically unmounted.

NFS Automounter Advantages

The automounter's major advantage is when combined with NIS, it provides a centralized, transparent view of the network. Other advantages are:

- Mounting done on an as needed basis. Automount unmounts all directories or files that are not in use. This reduces the risk of being hung if the system crashes and frees up dependencies on servers that are not currently in use.
- Load balancing. Automount finds the nearest server for replicated, read-only files by going through the fewest bridges and routers to handle mount requests. By distributing the client workload, automount helps to reduce the load of the most heavily used network hardware.

NFS Automounter Disadvantages

Automount has the following disadvantages:

- If you have a lot of directories in your search path, logging into a system using automount can greatly increase log-in time. This can be avoided by not listing the directories in your search path and create shell scripts for those utilities you actively use and put them in `/usr/local/bin`. This will cause automount to mount the directories or files or directories when needed rather than when you log in.
- When a directory to be mounted is on a local host, automount makes a symbolic link to the UNIX file mount point instead of remounting the directory on the local host. When moving down into a directory that was automounted from the local host, the directory shown by `pwd` will vary depending upon whether you arrived in the directory using the local host's path to the file or the automount directory. To eliminate confusion, use a shell script to replace the `pwd` command. The alias uses the directory that gets the current directory, leaving the symbolic link names alone. Adding this alias to each user's start up shell will hide the confusing directories produced by automount.

How Automount Works

When started, automount consults a series of maps for a list of directories or files to mount. The names of the maps can be passed to automount from the command line or from an NIS master map. The automount daemon serves the mount points specified in the maps and looks like an NFS server to the kernel. It uses the maps to locate an NFS file server with the appropriate directory. When the NFS file server is located, automount mounts the directory in a directory that serves as a temporary location (`/tmp_mnt`) and replaces the directory entry for the directory or subdirectory with a symbolic link to the temporary location. The `/tmp_mnt` directory is automatically created by automount.

Installation

The installation procedures for the HP 9000 Series 300/400 and Series 600/700/800 computers are slightly different. These differences will be noted in the sections that follow.

Key Terms

Term	Definition
Cluster	One or more workstations linked together with a local area network (LAN), and sharing a global directory attached to the root server. For more information on cluster concepts
Cluster Auxiliary Server	A cluster client with a disk drive that contains files shared by the other members of the cluster.
Cluster Node (Cnode)	Any node operating in an HP-UX cluster environment, including cluster clients and cluster servers.
Cluster Client	A node in an HP-UX cluster that uses networking capabilities to share directories or files, but does not have its root directory directly attached. For HP-UX 8.0, cluster clients can have locally mounted disks for local data storage.
Cluster Root Server	The only node in an HP-UX cluster that has the root directory directly attached to it.
Context Dependent File (CDF)	A hidden directory that contains all the versions of a file needed by the different cnodes.
Heterogeneous Cluster	A diskless cluster with more than one type of computer architecture (e.g., Series 300 and Series 800).

Term	Definition
Homogeneous Cluster	A diskless cluster composed of nodes of only one computer architecture (e.g., Series 300 only).
Internet Address	A four-byte quantity that is distinct from a link-level address and is the network address of a computer node. This address identifies both the specific network and the specific host on the network.
LAN	Local Area Network.
Network Information Service (NIS)	An optional network service composed of databases (maps) and processes that provide NIS clients access to the maps. NIS enables you to administer these databases from one node.
NFS	Network File System.
Node	A computer system that is attached to or is part of a computer network.
update	The HP-UX command that installs or updates software onto the system.

NFS Installation Checklist

The following steps are a checklist of NFS installation procedures. You may have already completed several of these steps. You will most likely start with Step 4. Steps 4 through 6 are explained in detail in this chapter.

1. Prepare your HP 9000 system for operation:
 - a. Inspect hardware.
 - b. Create and maintain a network map.
2. Ensure your computer is running either the LAN/9000 software (refer to the *Installing and Administering LAN/9000* manual), the FDDI/9000 software (refer to the *Installing and Administering FDDI/9000 Software* manual), or the Token Ring/9000 software (refer to the *Installing and Administering Token Ring/9000* manual).

Ensure that your computer's HP-UX operating system, your LAN/9000, FDDI/9000, or Token Ring/9000 software, and the NFS software that you are about to install all have the same version number. If you do not know which version of HP-UX your computer is running, execute the `uname -r` command.

If the versions do not match, run `update` to install the correct HP-UX operating system version. Refer to the *System Administration Tasks* manual for information on the update procedure.

3. Install the NFS software. You will need to use the update program to install the NFS software. Refer to the *System Administration Tasks* manual for detailed update information. Do the following:
 - a. Use the `/etc/update` command.
4. Add your HP 9000 computer to the network using your LAN/9000, FDDI/9000, or Token Ring/9000 software. Refer to *Installing and Administering LAN/9000*, *Installing and Administering FDDI/9000 Software*, and the *Installing and Administering Token Ring/9000* manuals. Do the following:
 - a. Assign an internet address.
 - b. Edit `/etc/rc`, `/etc/netlinkrc`, and `/etc/netnfsrc` manually or use SAM (System Administration Manager).
 - c. Verify that device files exist for the node's LAN, FDDI, Token Ring software; if they do not, you must create them.
 - d. Reboot the system.

Prepare the HP 9000 System

To prepare your HP 9000 computer for operation on LAN, FDDI, or Token Ring you must ensure your hardware is installed correctly.

For LAN hardware installation instructions for your computer, refer to the following documentation:

- *LAN Interface Controller (LANIC) Installation and Reference Manual.*
- *Twisted-Pair MAU Installation Guide.*
- *LAN Cable and Accessories Installation Manual.*
- *Installing and Administering LAN/9000.*

Another step in preparing your system is to update your network map with all new installation information (e.g., new computers, cable changes). If you do not have a network map, HP strongly recommends you create one. Refer to each of the *Installing and Administering LAN, FDDI, and Token Ring/9000* manuals for guidelines.

For FDDI/9000 hardware installation instructions for your computer, refer to the following documentation:

- *FDDI Adapter Installation Guide.*
- *FDDI Quick Start.*
- *Installing and Administering FDDI/9000 Software.*

For Token Ring/9000 hardware installation instructions for your computer, refer to the following documentation:

- *Token Ring Adapter Installation Guide.*
- *Token Ring Quick Installation Guide.*
- *Installing and Administering Token Ring/9000.*

Install the NFS Software

Before installing NFS Services software, you should ensure the following items are true:

- Your computer's HP-UX operating system, your LAN, FDDI, or Token Ring/9000 software, and your NFS software all have the same version number. Otherwise, the network may malfunction. To check which version of HP-UX you are currently running, execute the `uname -r` command.
- The LAN, FDDI, or Token Ring/9000 software is installed. To verify whether the software has been installed, check with your systems administrator. If you are the systems administrator, and you have not already installed the software, refer to the *Installing and Administering LAN, FDDI, or Token Ring/9000* manuals for installation and configuration instructions.

Use update Program

Before installing NFS Services, refer to the *System Administration Tasks* manual to familiarize yourself with the update program's menu operations and device file information.

After you are certain the required HP-UX and LAN, FDDI, or Token Ring/9000 software is installed, use the `/etc/update` program to install the NFS Services software. The `/etc/update` program takes you through the installation procedure step by step.

After you finish installing the NFS software, log in as superuser and display the `/etc/newconfig` directory. The installation added the following files to the `/etc/newconfig` directory. You will use these files when you configure the NFS Services, Network Information Service, and Virtual Home Environment:

- `/etc/newconfig/netgroup`
- `/etc/newconfig/yp_Makefile`
- `/etc/newconfig/netnfsrc`
- `/etc/newconfig/ypmake`
- `/etc/newconfig/rpc`
- `/etc/newconfig/netnfsrc2`
- `/etc/newconfig/ypxfr_1perday`
- `/etc/newconfig/vhe_mounter`
- `/etc/newconfig/ypxfr_1perhour`
- `/etc/newconfig/vhe_u_mnt`
- `/etc/newconfig/ypxfr_2perday`
- `/etc/newconfig/vhe_list`
- `/etc/newconfig/ypinit`

The configuration procedures are described later in this manual. For descriptions of these files and other files in `/etc/newconfig`, refer to `/etc/newconfig/README`.

Note If you have just updated previously existing NFS Services on a Series 300/400 system, then you have completed installing the NFS Services product. If you have just updated a Series 300/400 system to add NFS Services for the first time, you must now configure a new kernel to include NFS. To configure a new kernel, refer to the next section.

On a Series 600/700/800 system, you have completed installing the NFS services product, and you do not need to configure a new kernel unless the update program failed to generate a new kernel. To configure a new kernel, refer to the next section.

Configure a New Kernel

To prepare the NFS Services product for use, you must configure a new HP-UX operating system kernel if you are installing NFS Services for the first time.

If your kernel was constructed from the standard kernel file (`/etc/conf/dfile` on a Series 300/400 or `/etc/conf/gen/S800` on a Series 600/700/800), you can use SAM (System Administration Manager) to configure a new kernel that includes NFS.

If your kernel is based upon a customized kernel file, you must manually configure a new kernel. On a Series 300/400, your kernel file must contain the uncommented entry:

```
nfs
```

On a Series 600/700/800, your kernel file must contain the uncommented entry:

```
include nfs;
```

Note If you are configuring NFS in an HP-UX cluster environment, you must configure NFS into the kernel on all cnodes in the cluster.

See the *System Administration Tasks* manual for instructions on configuring a new kernel.

Add a Computer to the Network

If you have not already done so, refer to the *Installing and Administering LAN*, *Installing and Administering FDDI/9000 Software*, or *Installing and Administering Token Ring/9000 Software* manuals for instructions on adding your HP 9000 computer to the network. You will need to perform the following steps:

1. Determine and assign an internet address.
2. Edit `/etc/rc` and `/etc/netlinkrc` manually or use SAM (System Administration Manager).
3. Verify that device files exist for the node's LAN, FDDI, or Token Ring software; if they do not, create them.

After rebooting the system, log in as superuser and refer to the “NFS Configuration and Maintenance,” “NIS Configuration and Maintenance,” and “VHE Configuration and Maintenance” chapters to configure your system with NFS, NIS (if applicable), and VHE. Refer to the “Configuring and Maintaining the BIND Name Server” chapter in the *Installing and Administering ARPA Services* manual to configure the BIND Name Server if applicable.

NFS Configuration and Maintenance

This chapter describes basic NFS configuration without the Network Information Service (NIS). The latter portion describes how to administer and maintain the NFS service once you have it configured. For specific NFS information, refer to the following sections:

- Key terms.
- Guidelines.
- NFS configuration.
- NFS maintenance.

Note All references to servers and clients in this chapter apply to NFS servers and NFS clients unless otherwise specified.

Key Terms

Term	Definition
Alias	A term for referencing alternate networks, hosts, and protocol names.
Client	- A node that requests data or services from other nodes (servers). - A process that requests other processes to perform operations. <i>Note: An NFS client can also be configured as any combination of an NFS server, NIS client, or NIS server. (An NIS server <i>must</i> also be configured as an NIS client.)</i>
Clock Skew	A difference in clock times between systems.
Cluster	One or more workstations linked together with a local area network (LAN)
Cluster Auxiliary Server	A cluster client with a disk drive that contains files shared by the other members of the cluster.
Cluster Client	A node in an HP-UX cluster that uses networking capabilities to share directories or files, but does not have its root directory directly attached. For HP-UX 8.0, cluster clients can have locally mounted disks for local data storage.
Cluster Node (Cnode)	Any node operating in an HP-UX cluster environment, including cluster clients and cluster servers.
Cluster Root Server	The only node in an HP-UX cluster that has the root directory directly attached to it.
Daemon	Background programs that are always running, waiting for a request to perform a task.
Export	To make a directory available to remote nodes via NFS.
File System	An entire unit (disk partition) that has a fixed size.
GID	A value that identifies a group in HP-UX.
Hard Mount	A mount that causes NFS to retry a remote directory request until it succeeds, you interrupt it (default option), or you reboot the system.

Term	Definition
Host	A node that has primary functions other than switching data for the network.
Import	To obtain access to a remote directory from an outside source; to mount a remote directory.
Internet Address	A four-byte quantity that is distinct from a link-level address and is the network address of a computer node. This address identifies both the specific network and the specific host on the network.
Interruptable Mount	A mount that allows you to interrupt an NFS request by pressing an interrupt key. (Though the interrupt key is not standardized, common ones include [CTRL]-[C] and [BREAK].)
Locally Mounted File System	A directory that is locally mounted on a cluster client in a diskless cluster.
Mount	To obtain access to a remote or local directory or directory (import).
Mount Point	The name of the directory on which a directory is mounted.
Netgroup	A network-wide group of nodes and users defined in /etc/netgroup.
Network Information Service (NIS)	<p>An optional network service composed of databases (maps) and processes that provide NIS clients access to the maps. NIS enables you to administer these databases from one node.</p> <p>NIS may or may not be active; check with your system administrator.</p>
NFS	Network File System.
NIS Domain	A logical grouping of NIS maps (databases) stored in one location. NIS domains are specific to NIS and are not associated with other network domains.
Node	A computer system that is attached to or is part of a computer network.

Term	Definition
Server	<p>- A node that provides data or services to other nodes (clients) on the network.</p> <p>- A process that performs operations as requested by other processes.</p> <p><i>Note: An NFS server can also be configured as any combination of an NFS client, NIS client, or NIS server. (An NIS server <i>must</i> also be configured as an NIS client.)</i></p>
Soft Mount	An optional mount that causes access to remote directories or files to abort requests after one NFS attempt.
UID	A value that identifies a user in HP-UX.
Unmount	To remove access rights to a directory or disk that was mounted via the mount command.
update	The HP-UX command that installs software onto the system.

Guidelines

Refer to the following guidelines for information regarding:

- Network memory.
- Configuration files.
- Daemons.
- Servers.

Network Memory

Network memory is configurable using three parameters: `netmeminit`, `netmemmax` and `netmemthresh`. The default values are generally sufficient for most NFS configurations. However, if you change these parameters, do not set `netmemmax` equal to or less than `netmemthresh`.

To check the memory available on your network, enter the following command:

```
netstat -m
```

For more information about network memory configurations, refer to the *System Administration Tasks* manual.

Configuration Files

The following table lists the files that must be configured (unless otherwise stated) for your system to operate correctly. (Refer to the *HP-UX Reference Guide* for detailed information.)

Configuration File	Description
<code>/etc/checklist</code>	Contains a list of directories or files that are automatically mounted at boot time.
<code>/etc/exports</code>	Contains a list of directories or files that clients may import. <i>Note:</i> Create this file only on servers.
<code>/etc/inetd.conf</code>	Contains information about servers started by <code>inetd</code>
<code>/etc/netgroup</code>	Contains a mapping of network group names (<code>netgroups</code>) to a set of node, user, and NIS domain names; both <code>/etc/exports</code> and <code>/etc/passwd</code> can use the <code>netgroups</code> defined in <code>/etc/netgroup</code> . Classifies the nodes for remote mounts. For <i>ARPA Services</i> , classifies the users for remote logins and remote shells. You can specify <code>netgroups</code> in <code>/etc/hosts.equiv</code> and <code>\$/HOME/.rhosts</code> . Configuring this file is optional.
<code>/etc/netnfsrc</code>	Automatically executed at boot time to start the NFS networking (e.g., starts daemons and servers, defines servers and clients.) <i>Note:</i> For 8.0, <code>etc/netnfsrc</code> has changed significantly. A new <code>etc/netnfsrc</code> file is loaded during the install and update procedure. Lines in which configuration variables are set are propagated from the old version to the new version. The old version is saved in <code>etc/netnfsrc.OLD</code> . You must copy any customization necessary from the old version. In an HP-UX environment
<code>/etc/netnfsrc2</code>	Perform mount operations for all NFS mount entries found in <code>/etc/checklist</code> . This file is static; it is already correctly configured.

Configuration File	Description
<code>/etc/rpc</code>	<p>Maps the RPC program names to the RPC program numbers and vice versa.</p> <p>This file is static; it is already correctly configured.</p>
<code>/usr/adm/inetd.sec</code>	<p>Checks the internet address of the host requesting a service against the list of hosts allowed to use the service.</p> <p>Specifies how many remote users can simultaneously start remote services in the local system and which remote hosts (or networks) can use the system.</p>

Daemons

The following table lists the networking daemons (background programs) that are always running, waiting for a request to perform a task. The parenthetical comments refer to the *HP-UX Reference* sections where you can go for more information.

Daemon	Description	
<i>biod</i> (1M)	Asynchronous block I/O daemons for NFS clients.	
<i>inetd</i> (1M)	Internet daemon that listens on service ports. It: <ul style="list-style-type: none">- Reads <i>/etc/inetd.conf</i> to determine the appropriate server for handling the incoming request.- Listens for and accepts network requests.- Invokes the appropriate server. <p><i>Note:</i> Since <i>inetd</i> contacts <i>portmap</i> on behalf of the servers it starts, you must start <i>portmap</i> before starting <i>inetd</i>.</p>	
<i>nfsd</i> (1M)	NFS server daemon that responds to client directory requests. When a client program needs to read or write in a remote directory, it sends a request to that system's <i>nfsd</i> process.	If operating in an HP-UX cluster environment, <i>nfsd</i> must be running on any cnode with a local directory that will be exported via NFS. Any <i>nfsd</i> daemons running on cnodes without locally mounted directories or files are ignored by the cnodes.
<i>pcnfsd</i> (1M)	Daemon that authenticates a PC user's access to files. It takes the user name and password, and then does <i>one</i> of the following: <ul style="list-style-type: none">- Succeeds (returns a valid UID and GID).- Fails (indicates the name and password are unacceptable). <p><i>Note:</i> Though <i>pcnfsd</i> enables PC users to use printer spooling facilities on HP-UX systems, they <i>must</i> have the appropriate PC networking software product for it to work.</p>	

Daemon	Description
<i>portmap(1M)</i>	<p>Daemon that converts RPC program numbers into port numbers. When <i>inetd</i> starts, it tells <i>portmap</i>:</p> <ul style="list-style-type: none">- Which RPC servers it is listening for.- On which ports it is listening.- The RPC program numbers and versions it serves. <p>When a client makes an RPC call to a given program number, it first contacts <i>portmap</i> on the server node to determine the port number where RPC requests should be sent.</p> <p><i>Note:</i> Since <i>inetd</i> contacts <i>portmap</i> on behalf of the servers it starts, you <i>must</i> start <i>portmap</i> before starting <i>inetd</i>.</p>

Servers

The following table lists the networking servers (processes that perform operations as requested by other processes). The parenthetical comments refer to the *HP-UX Reference* sections where you can go for more information.

Server	Description
<i>mountd</i> (1M)	<p>Answers directory mount requests by reading <i>/etc/xtab</i> to determine which directories or files are available to nodes and users; invoked by <i>inetd</i>.</p> <p>The <i>showmount</i> command calls <i>rpc.mountd</i> to list the clients with local directories or files mounted.</p> <p>If operating in an HP-UX cluster environment, <i>mountd</i> must be running on any cluster <i>cnode</i> that wishes to export its local directory via NFS. The <i>mountd</i> servers are ignored on any cluster client that does not have locally mounted directories or files.</p>
<i>rstatd</i> (1M)	<p>Returns statistics obtained from the kernel; invoked by <i>inetd</i>.</p> <p>The <i>rup</i> program uses <i>rpc.rstatd</i>.</p>
<i>rusersd</i> (1M)	<p>Lists the users on the local host; invoked by <i>inetd</i>.</p> <p>The <i>rpc.rusersd</i> server provides the <i>rusers</i> program information about the local users. The <i>rusers</i> program then sums and displays the information.</p>
<i>rwalld</i> (1M)	<p>Handles all <i>rwall</i> requests; invoked by <i>inetd</i>.</p> <p>The RPC program <i>rwall</i> sends a message to <i>rpc.rwalld</i> on a given host. Each <i>rpc.rwalld</i> accepts this message and writes it to all users on the host it is serving using <i>wall</i>.</p>
<i>sprayd</i> (1M)	<p>Records the packets sent by <i>spray</i>; invoked by <i>inetd</i>.</p>

Note These processes are contained in the library */usr/include/librpcsvc.a*. Applications that call these processes must be linked to this library. For example:
`cc program.c -librpcsvc`

NFS Configuration

Configuring your system is the process of setting up your software so it operates correctly according to your specifications. The following sections describe the steps you must perform to configure NFS Services on nodes that reside on your network. You can perform some NFS Services configurations in SAM (System Administration Manager), a tool that automates the configuration process. Go to the following sections for detailed configuration instructions (notice that both the SAM and manual configuration methods are included here):

- Compare the files in the `/etc/newconfig` directory to their corresponding existing files.
- Set UIDs and GIDs.
- Create an NFS server and an NFS client using SAM.
- Create an NFS server manually (without SAM).
- Create an NFS client manually (without SAM).
- If applicable, configure the Network Information Service (NIS). (Refer to the NIS Configuration and Maintenance chapter.)
- If applicable, configure the Virtual Home Environment (VHE) service. (Refer to VHE Configuration and Maintenance chapter.)
- Execute `/etc/netnfsrc` (or reboot) when you are finished with all of the configuration, including setting up NIS and VHE.

Compare `/etc/newconfig` files to existing files

When you installed the NFS Services software, several new files were copied into the `/etc/newconfig` directory. Perform the following steps to prepare the NFS Service for configuration:

1. Compare each `/etc/newconfig` file listed below with its counterpart shown in the following list.

File in /etc/newconfig directory	Counterpart in /etc directory
netgroup	netgroup
netnfsrc	netnfsrc
rpc	rpc
netnfsrc2	netnfsrc2

2. If the files are the same, then skip to the next section, Set UIDs and GIDs.
3. If you have previously customized the files that exist in the /etc directory or if the files are from an older version of the software, they will differ from those in /etc/newconfig. If there are differences, copy the current files in /etc to a safe location and do *one* of the following:
 - Change the versions in /etc to reflect the differences in the files in /etc/newconfig.
 - Copy the files in /etc/newconfig to /etc. Then customize the files in /etc if necessary.

Set UIDs and GIDs

The UID field from an /etc/passwd entry and the GID field from an /etc/group entry authenticate NFS users. The client passes this UID and GID to a server for use when checking file ownership and permission.

To ensure only the users in the correct group receive the privileges set by the file's owner, edit /etc/passwd and /etc/group so that each user has one unique UID and one unique GID that is the same on all servers and clients.

If you are using the Network Information Service (NIS), you can configure NIS so you can centrally administer `/etc/passwd` and `/etc/group`. (Local UIDs and GIDs are not required if you are using NIS.)

If you are not using NIS, you can use *one* of the following two methods to either create new `/etc/passwd` and `/etc/group` files or modify the existing ones:

- Create one `/etc/passwd` and one `/etc/group` file to ensure UIDs and GIDs are consistent for each NFS user across the network. Copy these files to all NFS network nodes.

When updating UIDs or GIDs, you will need to recopy the files to each node. You can automate this process by using shell scripts and the ARPA Services.

A disadvantage of this method is that it gives exactly the same access to all users across the network. A user with a valid password for a superuser account would have superuser privileges on all nodes configured in this fashion.

- Edit `/etc/passwd` and `/etc/group` on each node to ensure UIDs and GIDs are consistent for each user across the network.

If you modify UIDs or GIDs affecting more than one node, you will have to modify each node affected by the change. For example, if adding a new user you will need to update the `/etc/passwd` and `/etc/group` files residing on each system to which the new user will have access.

Though more time consuming and error prone, this method allows each system to have a different set of users.

Create an NFS Server and an NFS Client Using SAM

SAM (System Administration Manager) provides an automatic method for configuring your local system to be an NFS server or NFS client. You must be a superuser to use SAM.

Tips for using SAM

Remember the following tips when you use SAM. You can also get more information about SAM by activate the **Help** button.

- Use your mouse or your keyboard's cursor control to navigate through your configuration.
- You can select menu items using either one of the following methods:
 - Use your mouse and highlight the menu item you want and activate the **OPEN** button.
 - Move your cursor to the menu item you want using Tab or the arrow keys and press return.
- The "List", "View", "Options", and "Actions" menus allow you to customize the screens where actions like add, delete, modify, etc. are performed and enable you to select an action to perform on one or more items.
- You can activate the online help by either pressing the F1 key or highlight and activate the **Help** button.
- After filling in or choosing options for configuration, you are presented with three buttons to choose from. They are **Apply**, **Cancel**, and **OK**. The **Apply** button will perform the task (accept your configured parameters) and leave you in the current menu. The **Cancel** button will exit you out of the current menu into the previous menu. The **OK** button will perform your task and return you to the previous menu after a message is displayed.

Note You cannot configure `/etc/netgroup` in SAM. If you wish to edit this file, go to the section, "Create an NFS Server Manually."

Move to the NFS Configuration Menu

All NFS configurations available in SAM are done in the *Networked File Systems (NFS)* window. This section explains how to move to that window where you can select the task you wish to perform.

1. At the HP-UX prompt, type: `sam`

2. At the SAM Main Window, highlight *Networking/Communications* and activate the **OPEN** button.

Note Make sure you have already installed and configured your LAN, Token Ring, or FDDI links and that you have configured your system to communicate with other systems before you use SAM to configure NFS. Refer to the following documentation for more information on configuration of LAN, Token Ring, or FDDI links and network connectivity: *Installing and Administering LAN/9000 Software*, *Installing and Administering Token Ring/9000 Software*, and *Installing and Administering FDDI/9000*.

Your system may not be able to communicate with other systems until you do the following:

Note the following information before you begin:

If your system is configured to use the NFS Network Information Services (NIS) or the ARPA Service's BIND Name Service for hostname to address mapping, you cannot use SAM to add NFS Services connectivity information about a remote system. The *System-to-System Connectivity* window edits only the `/etc/hosts` file; it does not edit an NIS or BIND Name Service database.

If you must go through a gateway to add connectivity information to reach the remote system, SAM will prompt you for the gateway's hostname and IP address. With this information, SAM will automatically configure the necessary routing by executing an `/etc/route add_host` command and adding it to `/etc/netlinkrc`.

If there is just one gateway you use to reach all systems on other parts of the network, select **Use current default gateway**, in the *Add Internet Connectivity* window.

Information you will need to complete this task includes:

- Official host name of the remote system.
- IP Address of the remote system.
- Alias names (optional). An alias name is required if more than one card has been installed in the system.

To perform configuration of network connectivity:

1. Highlight *System-to-System Connectivity* and activate the **OPEN** button.

2. To enable your system to communicate with other systems using the TCP/IP protocol, highlight *Internet Connectivity* and activate the **OPEN** button. The object list will display the remote system names and IP addresses that are already configured.
3. To add remote system entries, choose **Add** from the "Actions" menu and then enter the information about the system you want your system to connect with.
4. When you are finished entering remote system information, activate the **OK** button to add the system to your system's `/etc/hosts` file and return to the *System-to-System* Object List (activate the **APPLY** button if you are configuring more than one system).
5. SAM updates the list of remote systems to include the remote system you configured.

Note You can modify or remove remote systems and modify default gateways by highlighting the Remote System Name from the list of remote systems in the *System-to-System Connectivity* window and choosing **Modify**, **Remove**, or **Modify Default Gateway** from the "Actions" menu.

6. To exit SAM, choose **Exit** from the "List" menu to return to the *System-to-System Connectivity* window, and then activate the **Exit SAM** button.
7. To verify remote system configuration:
 - a. View the list of remote systems you can communicate with using a symbolic name by typing the following command at the HP-UX prompt:

`more /etc/hosts`
 - b. View the configured destination reached through gateways and the gateways used to reach those destinations by typing the following command at the HP-UX prompt:

`netstat -r`
 - c. If the `r1b(1M)` remote communications diagnostic fails, check the following:
 - i. Check the node name on the local and remote systems. If the `.domain.organization` parts of the name are different, be sure to specify a fully-qualified node name for the remote system.
 - ii. If you are using gateways, probe proxy must be setup. Refer to the *Node Manager's Guide* for more information.

- d. To view information about connectivity with a remote system, execute the following command:

```
arp 191.2.1.2
```

The output of the command is the ARP cache entry, which show the mapping between IP address and station address. If the station address is incomplete, the problem may be because this system has not received an ARP response from the remote system.

Note If the route to the remote system is through an indirect route, marked G in `netstat -r` output, the IP address of the gateway, not the remote system, will appear in the ARP cache.

To configure an NFS client using SAM, perform the following:

1. At the SAM Main window, highlight Networking/Communications and activate the **OPEN** button.
2. At the Networking/Communications window, highlight *Networked File Systems (NFS)* and activate the **OPEN** button.
3. Highlight *Remote File Systems Mounted* and activate the **OPEN** button. This edits your `/etc/netnfsrc` file.
4. To add remote directories to be mounted, choose **Add** from the "Actions" menu and then enter the information about the remote directory you want to mount to.
 - a. Enter the Remote System Name. This is the system name where the directory or file you wish to access resides.
 - b. Enter the Remote Directory Name. This is the directory name of the directory you wish to access.
 - c. Enter the Local Directory Name. This is the local directory where you want the remote directory to be mounted.
 - d. Choose when to mount. You have two mount options to choose from, they are: **Now** and **On Boot**. If **Now** is chosen, the directory will be available when the information in this window has been accepted. If **On Boot** is chosen, the directory will be available when this system is booted.

- e. Choose write protection. Write protection provides the type of access the user will have on the remote directory. There are two options to choose from: Read-Only, meaning the user can only read the information in the directory and Read/Write, meaning the user can read and write to the directory.
- f. Set userID Execution. This indicates whether the user must have a user ID when trying to access the remote directory. Choosing Yes means a user ID is not required.
- g. Change Default Mount Options. This takes you into the *NFS Mount Options* window. You can change the following mount options:
 - i. Choose one of the mount options. There are two choices: The first choice is Hard, hard mounted directories or files cause NFS to retry a request until the server responds. If a server does not respond to a hard mount request, NFS writes the following message in the network error log file: NFS: server host_name not responding, still trying. The second choice is Soft, soft mounted directories or files abort requests after one attempt. NFS writes an error to the log file if the server does not respond to a request: NFS server host_name not responding, giving up.
 - ii. Read buffer size (rsize). This specifies the maximum read request size used in communicating with the server.
 - iii. Write buffer size (wsize). This specifies the maximum write request size used in communicating with the server.
5. When you are finished entering mount information, activate the **OK** button to perform the task and return to the Add Remote Directory window.
6. When you are finished entering the remote directory information, activate the **OK** button to add this remote directory information to your `/etc/hosts` file.
7. In the Remote File Systems Mounted window, you can enable the NFS client by choosing **Enable NFS client** from the "Actions" menu. You should see the remote directory information displayed in the window.

Go to the next section to determine which remote systems can have NFS access to your local directories or files.

To configure an NFS Server using SAM, do the following

1. At the Networking/Communications window, highlight *Networked File Systems (NFS)* and activate the **OPEN** button.
2. Highlight *Local File Systems Exported* and activate the **OPEN** button. This edits your `/etc/exports` file.
3. To add local directories that can be exported to remote systems, choose **Add** from the "Actions" menu and then enter the information about the directory you want to export to the remote systems. This will take you into the Add File System to Export window.
 - a. Enter the Local Directory Name. This the directory that will be exported to remote systems.
 - b. Enter an anonymous userid ID (anon). You can map anonymous, or unknown, user requests to uid. By setting the anonymous user ID in `/etc/exports`, the unknown user in an anonymous request is mapped to a well-known local user. If the anonymous user is mapped to nobody (the default), anonymous requests are accepted but have very few permissions to access files on the server. The information you need to complete this task are the remote system names to which you are allowing or denying access. If you specify uid for unknown user, you will be prompted for a login name.
 - c. Select asynchronous writes. Yes means that asynchronous write will be done. No means no asynchronous writes will be done.
 - d. Specify read only access. Selecting read only will cause the *Read Only Access* window to be displayed. The window enables you to add, modify or remove remote systems to have read only access to your local directory. When you have finished filling in the information in this window, activate the **OK** button.
 - e. Specify read/write access. Selecting read/write will cause the *Read/Write Access* window to be displayed. This window enables you to add, modify or remove remote systems to have read/write access to your local directory. When you have finished, activate the **OK** button.
4. In the Local File Systems Exported window, you can enable the NFS server by choosing **Enable NFS server** from the "Actions" menu. You should see the local directories displayed and the directories access information displayed in the window.

Create an NFS Server Manually (Without SAM)

You must be superuser to create an NFS server. To create an NFS server, complete the following steps. These steps are described in detail in the sections that follow.

1. Edit `/etc/netnfsrc`.
2. Edit `/etc/inetd.conf`.
3. Edit `/usr/adm/inetd.sec` (if necessary).
4. Edit `/etc/hosts`.
5. Edit `/etc/netgroup` (optional).
6. Create and Edit `/etc/exports`.
7. Reboot the system (if necessary) or run `/exports -a`.

An NFS server can also be configured as any combination of an NFS client, NIS client, or NIS server. (An NIS server *must* also be configured as an NIS client.)

Note If you are configuring NFS in an HP-UX cluster environment, you must configure NFS into the kernel on all cnodes in the cluster. See "Configure a New Kernel" in Chapter 3.

1. Edit `/etc/netnfsrc`

The `/etc/netnfsrc` file activates the NFS daemons and servers.

- To define the node as an NFS server, set the `NFS_SERVER` variable to any digit other than zero.
- Set `START_MOUNTD` to any digit other than 0.
- `mountd` can be started from `netnfsrc` on `inetd`. If `mountd` has an entry in `inetdrconf` the `START_MOUNTD` should be 0.
- If the node is also a client, you may want to set the `NFS_CLIENT` variable to any digit other than zero now. (Refer to the "Create an NFS Client Manually" section to complete client configuration procedures.)
- If the node is also a server for PC-NFS requests, set the `PCNFS_SERVER` variable to any digit other than zero.

Client Only	NFS_CLIENT = 1 NFS_SERVER = 0
Server Only	NFS_CLIENT = 0 NFS_SERVER = 1
Both Client and Server	NFS_CLIENT = 1 NFS_SERVER = 1
Neither Client nor Server	NFS_CLIENT = 0 NFS_SERVER = 0
PC-NFS Server	PCNFS_SERVER = 1

You can refer directly to the comments (lines beginning with pound signs) for editing instructions and for descriptions of each activity executed by `/etc/netnfsrc`.

Note If you edit this file other than specified in this document, HP recommends you incorporate personal comments for future system administration.

```

#!/bin/sh
# netnfsrc      NFS startup file
##
# Depending on the configuration parameters you set within,
# this script sets up some or all of the following:
#* NIS specific:
#   domainname    the NIS domain name
#
# and starts up some or all of the following programs:
#   portmap      RPC (program_#,version) -> port_# mapper
#   nfsd         NFS daemons
#   biod        async BIO daemons
#   pcnfsd      PC-NFS daemon
#* NIS specific:
#   ypbind      NIS client process (all NIS nodes)
#   ypserv      NIS server process (NIS server only)
#   yppasswdd   NIS password daemon (NIS master server only)
##
# NFS_CLIENT    1 if this node is an NFS client, 0 if not
# NFS_SERVER    1 if this node is an NFS server, 0 if not
# Note: it is possible for one host to be a client, a server, both
#         or neither! This system is an NFS client if you will be
#         NFS mounting remote directories or files; this system is a server
#         if you will be exporting directories or files to remote hosts.
# See Also: nfsd(1M), mount(1M)
##
NFS_CLIENT=0
NFS_SERVER=0
START_MOUNTD=0
.
.
.
PCNFS_SERVER=0

```

Allow or Deny access to specific RPC services (servers) using SAM

This task lets you allow or deny access to specific RPC services (servers). When you perform this task, you are editing the `/usr/adm/inetd.sec` file. The information you need to complete this task are the remote system names to which you are allowing or denying access. To perform this task:

1. At the Networking/Communications window, highlight *Security* and activate the **OPEN** button.
2. Highlight *Internet Service* and activate the **OPEN** button.
3. To modify RPC services, highlight the RPC Services, you want to modify and you can choose to modify all Internet Services from the "Actions" menu. This will take you into the Modify Internet Security window. Select the system permissions you would like your RPC services to have.
4. When you have completed your task, activate the **OK** button.

2. Edit `/etc/inetd.conf` manually

To activate the RPC services, remove all `#` comment marks (pound signs) from `/etc/inetd.conf` lines beginning with `#rpc`. If you want one of these services activated but the line was removed, you may need to obtain a new version of `/etc/inetd.conf` from `/etc/newconfig`.

Note After editing `/etc/inetd.conf`, you must reconfigure `inetd` by entering:

`/etc/inetd -c`

RPC Services Security

The `inetd` security facility works only when the `inetd` executes a server. For the RPC services that do not exit after each service request, `inetd` provides a security check only for the first request. Successive requests bypass the `inetd` and are subject only to the security checking performed by the individual RPC services. However, you can make the `inetd` perform a security check for every RPC request by doing *both* of the following steps:

- Add the `-e` option to the `/etc/inetd.conf` entry for the RPC service.
- Specify the RPC service in the first field of `/usr/adm/inetd.sec`. (Refer to the next section, Edit `/usr/adm/inetd.sec`.)

Note Adding the `-e` option makes the RPC server respond slower since it has to restart for each request.

For information on C2 Security, refer to the *HP-UX System Security Manual* and the *HP-UX Beginner's Guide*.

RPC Entries

Refer to the following list for a brief description of each RPC service line present in `/etc/inetd.conf`.

```
rpc dgram udp wait root /usr/etc/rpc.mountd 100005 1 rpc.mountd -e
```

The `rpc.mountd` program is the server for the `mount` command and reads `/etc/exports` to see what the available directories or files are and to whom they are exported. It also keeps a list of all mounted directories or files.

The `-e` option forces `inetd` to perform a security check for `rpc.mountd` on every request.

```
rpc stream tcp nowait root /usr/etc/rpc.rexd 100017 1 rpc.rexd
```

The `rpc.rexd` program is the server for the `on` program. The program supports version 1.

```
rpc dgram udp wait root /usr/etc/rpc.rstatd 100001 1-3 rpc.rstatd
```

The `rpc.rstatd` program is the server for the `rup` command and provides kernel statistics. The program supports versions 1 through 3.

```
rpc dgram udp wait root /usr/etc/rpc.rusersd 100002 1-2 rpc.rusersd
```

The `rpc.rusersd` program is the server for the `rusers` command and provides information about active users on remote nodes and the amount of time they have been idle. The program supports versions 1 and 2.

```
rpc dgram udp wait root /usr/etc/rpc.rwalld 100008 1 rpc.rwalld
```

The `rpc.rwalld` program writes a message sent by `rwall` to all users logged on to the system. The program supports version 1.

```
rpc dgram udp wait root /usr/etc/rpc.sprayd 100012 1 rpc.sprayd
```

The `rpc.sprayd` program is the server for the `spray` command and accepts RPC requests, reads UDP packets, and then tells how fast it read them; you can use the results to gauge performance. The program supports version 1.

```
rpc dgram udp wait root /usr/etc/rpc.rquotad 100011 1 rpc.rquotad
```

The `rpc.rquotad` program is the server for the `quota` command. The daemon returns data regarding disk quotas for NFS mounted directories or files.

3. Edit `/usr/adm/inetd.sec` (if necessary)

NFS operates under the assumption you have a friendly network; meaning, you can trust all users attached to your network. Since this assumption may not apply to everyone, refer to the following sections to improve your file security.

The `/usr/adm/inetd.sec` configuration file is provided in the ARPA Services product. It is not solely for NFS access.

This file allows you to determine:

- How many remote services can run simultaneously on the local host.
- Which hosts are allowed to remotely use the local host.

Note If `inetd` is running, it rereads `/usr/adm/inetd.sec` after you make changes to it. Your changes apply only to services started after the file is reread, but not to any currently running services.

Set Maximum Number of Remote Connections

On the first line in `/usr/adm/inetd.sec`, enter the maximum number of simultaneous remote services to be started by `inetd` as shown in the following example:

```
MAXNUM number
```

If you do not specify a `MAXNUM` value, the default is 1000.

Specify Accesses to Services

Each entry in `/usr/adm/inetd.sec` has the following format (enter either `allow` or `deny`):

```
service_name allow/deny host_specifier(s)
```


/usr/adm/inetd.sec Entry Fields	Description
<i>service_name</i>	<p>Name of a valid service (including RPC services) with an entry in /etc/inetd.conf.</p> <ul style="list-style-type: none"> - For RPC services, <i>service_name</i> is the name of the service that matches its program number in /etc/rpc. This entry <i>must</i> have a corresponding entry in /etc/inetd.conf which contains the -e option. - Specify only one service per entry. - If an entry in /usr/adm/inetd.sec specifies the service name and nothing else, inetd allows all hosts to attempt access.
allow/deny	<p>The <i>allow</i> entry instructs inetd to approve the host or network for access to the specified service.</p> <p>The <i>deny</i> entry instructs inetd to disapprove the host or network for access to the specified service.</p>
<i>host_specifier(s)</i>	<p>Name of a host or a network listed in /etc/hosts or /etc/networks, or an internet address in the standard internet notation.</p> <ul style="list-style-type: none"> - You can specify more than one host or network by separating each <i>host_specifier</i> with a blank or tab. - You can use the * (wild card character) or - (range character) in any field of a network or host address. - You cannot use aliases.

RPC Services Security

You can make `inetd` perform its `inetd.sec` security check for every RPC request by following these two steps.

1. Add the `-e` option to the RPC service line in `/etc/inetd.conf`. (Refer to the 2. Edit `/etc/inetd.conf` section or `inetd.conf`.)

EXAMPLE: `rpc dgram udp wait root /usr/etc/rpc.mountd 100005 1 rpc.mountd -e`

2. Specify the RPC service in the first field in `/usr/adm/inetd.sec`.

<code>/usr/adm/inetd.sec</code> Example RPC Entry	Effect on System Security
<code>mountd allow hostA</code>	Allows only <code>hostA</code> to access <code>rpc.mountd</code>
<code>walld deny 111.56.78.9 10.*</code>	Denies access to <code>rpc.rwalld</code> from the following hosts: <ul style="list-style-type: none">- 111.56.78.9 (internet address)- all hosts that are part of network 10.*

4. Edit `/etc/hosts`

Caution If NIS is running, do not edit `/etc/hosts` on any node except the NIS master server; otherwise, local changes will not be propagated.

If you have ARPA Services and have configured the BIND name server, do not edit the `/etc/hosts` file. See the Maintaining Network and Domain Data Files section of the BIND Name Server chapter in *Installing and Administering ARPA Services*.

As node manager, you must configure this file for your host. You can add entries to this file either automatically with the System Administration Manager (SAM) or manually by editing the file. This section describes how to configure `/etc/hosts` manually.

Adding IP Addresses

If your host has more than one IP address (for multiple network interfaces), you must add entries for every IP address. These entries must have the same official host name but different

aliases or have different official hostnames. This is so that different IP addresses (network interfaces) are distinguished (and can be referenced) by different aliases or hostnames.

Note You can copy the official host data base maintained at the Network Information Control Center (NIC) for ARPA Internet networks. (Refer to the "Military Standards and Request for Comment Documents" section of Chapter 1 for information on how to contact the NIC.) Be sure to check the format of files received from the NIC.

If your host accesses a multi-homed host (one with more than one link interface), make sure the internet address for that host is correct in the `/etc/hosts` file with respect to your host. For example, in the networking scheme Figure 4-1, host paul and host barb access multi-homed host mickie via internet address 192.6.21.2. Hosts dean and dennis, on the other hand, access host mickie via internet address 192.6.36.3.

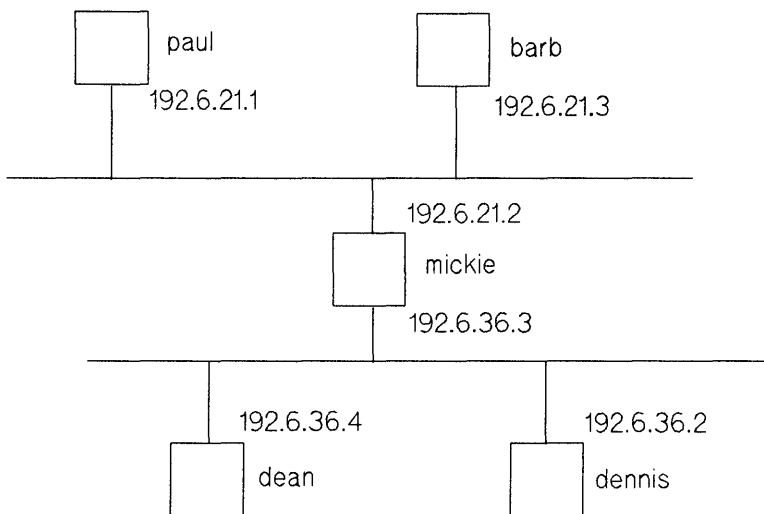


Figure 4-1. Multi-Homed Host Network Scheme

Syntax for `/etc/hosts`

Each host (including the local host) has a one line entry in the `/etc/hosts` file. Each entry in the `/etc/hosts` file takes the following form:

internet_address official_host_name [alias(es)]

<i>internet_address</i>	Network address that uniquely identifies the node. <i>Internet_address</i> must be in dot notation. Refer to the Assigning an Internet Address section in <i>Installing and Administering LAN</i> for more information on internet addresses.
<i>official_host_name</i>	Name of the node. Host names can contain any printable character except white spaces, newline, or the comment character (#). By convention, <i>official_host_name</i> should be the same as the system host name assigned with the HP-UX hostname command.
<i>alias(es)</i>	Common name or names for the node. An <i>alias</i> is a substitute for <i>official_host_name</i> . <i>Alias</i> names are optional and are not supported by all the commands that use <i>/etc/hosts</i> .

Format for /etc/hosts

- Lines cannot start with a white space (tabs or blanks).
- The fields can have any number of blanks or tab characters separating them.
- Comments are allowed and are designated by a "#" in front of the comment text.
- Trailing blanks and tab characters are allowed.
- Blank lines are allowed.

Example of /etc/hosts Entry

The /etc/hosts entry for a node with:

- The address 192.45.36.5.
- The official host name *hpdxsg*.
- The alias name *bullfrog*.

Looks like:

```
192.45.36.5 hpdxsg bullfrog
```

Permissions

The /etc/hosts file should be owned by user *root*, group *other* and have 0444 (r-r-r) permission.

Refer to the /etc/hosts file for examples of the actual format and contents. For more information on /etc/hosts, refer to the *hosts(4)* entry in the *HP-UX Reference* manual.

Verification

To view the list of remote systems you may communicate with, type the following command at the HP-UX prompt:

```
more /etc/hosts
```

To verify that /etc/hosts is being used to do host name to address mapping, use *nslookup* as described in the previous section, 3. Configure Host Name to Address Mapping.

To view the destinations reached through gateways and the gateways used to reach those destinations, type the following command at the HP-UX prompt:

```
netstat -r
```

The listing from this command may appear slowly, as it attempts to find the names associated with the network addresses used to perform routing.

Copying a Remote /etc/hosts File to Your Local Host

When you first configure your host's /etc/hosts file, it is very small. If you want to get a copy of a larger, more complete /etc/hosts file from another host, you can do it *one* of two ways:

- If you have ARPA Services on your host, go to the *Installing and Administering ARPA Services* manual and use the method described in the "Editing /etc/hosts" section of Chapter 2.
- If you have NFS Services ONLY, the method is more complicated and is described in the following example.

In the following example, your local host is named *myhost* and the remote host that has the complete /etc/hosts file is named *otherhost*. Perform the following steps:

1. Using either SAM or the manual method, add connectivity information about *otherhost* to the /etc/hosts file on *myhost*.
2. Using either SAM or the manual method, make *otherhost* an NFS server so that it allows *myhost* access to the *otherhost* root (/) directory.
3. Using either SAM or the manual method, make *myhost* an NFS client.
4. On *myhost*, mount the *otherhost* directory "/", copy /etc/hosts from *otherhost*, then unmount the *otherhost* "/" directory. See the following example:

```
mkdir /tmp/exmpl
mount otherhost:/ /tmp/exmpl
cp /tmp/exmpl/etc/hosts /etc/hosts
umount /tmp/exmpl
rmdir /tmp/exmpl
```

If you overwrite your local /etc/hosts file with a copy from another host, you may need to bring it up to date by adding unofficial aliases or unknown hosts, including your own host.

5. Edit /etc/netgroup

Caution If NIS is running, do not edit /etc/netgroup on any node except the NIS master server; otherwise, local changes will not be propagated.

The /etc/netgroup file enables you to define a specific network-wide group of nodes as a netgroup. You can then limit directory access by exporting directories or files (via /etc/exports and exportfs) to the netgroups defined.

The system uses /etc/netgroup to verify host names whenever clients perform remote mounts. (Refer to *netgroup(4)* in the *HP-UX Reference*.)

For *ARPA Services*, the system uses /etc/netgroup to verify users when clients perform remote logins or remote shells. (Refer to *hosts.equiv(4)* in the *HP-UX Reference*.)

Add a line with the following format for each netgroup you wish to define.

The entry may contain any number of netgroup names:

```
netgroup_name1 netgroup_name2 netgroup_name3 ...
```

But then you must define these netgroups within `/etc/netgroup`:

```
netgroup_name1 member1 member2 ...
```

You can use the following conventions when editing the `/etc/exports` file:

- The member *n* is equal to the triple (*host_name*, *user_name*, *NIS_domain_name*).
- You can assign more than one triple to a netgroup by enclosing each separate set within parentheses (*host_name*, *user_name*, *NIS_domain_name*).
- Leave any of these three fields empty to signify a wild card (i.e., blank fields match anything). For example, (*.,research*) matches all hosts and users in the *research* NIS domain.
- A - (dash) in any of these three fields means *match nothing*. For example, (*-,mike,graphs*) does not match any hosts, but it does match the user *mike* in the *graphs* NIS domain.
- Each *host_name* must have an entry in `/etc/hosts`.
- The *NIS_domain_name* is the name of the NIS domain to which you currently belong. To determine your current NIS domain name, execute the `domainname` command.

The commands using `/etc/netgroup` assume you are not looking for any NIS domain other than the one assigned on your node.

EXAMPLES:

/etc/netgroup Example Entry	The Netgroup Includes
netgroup1 (,,)	Everyone on the network.
netgroup2 (,darren,graphic)	The user darren on any host in the graphic NIS domain.
netgroup3 (node_7,,graphic)	Any user on the node_7 host in the graphic NIS domain.
netgroup4 (node_2,john,)	The user john on the node_2 host in any NIS domain.
netgroup5 (,andy,graphic) (node_1,mike,)	The user andy on any host in the graphic NIS domain and the user mike on the node_1 host in any NIS domain.
netgroup6 (-,annette,graphic)	The user annette in the graphic NIS domain, no host included.

6. Create and Edit /etc/exports

A system on your network becomes an NFS server when there are directories or files from that system to be exported to the network. The NFS server is controlled by the `exportfs`, `rpc.mountd`, and `nfsd` daemons. You make the directories or files and their access restrictions, if any, available by your entries in the server's `/etc/exports` file. When you boot up the NFS server the `/etc/rc.local` file will automatically run the `exportfs` daemon which looks up `/etc/exports` and makes the directories or files available for NFS clients to access. You can export and unexport directories or files after the server is up or change access permissions of the exported directory by using the `exportfs` daemon. The `exportfs` daemon can be run at anytime by the super-user to alter the list or characteristics of exported directories and filenames. It exports and unexports directories or files to NFS clients on an as needed basis.

- The server must have the directory mounted locally before it can be exported.
- Any directory in `/etc/exports` can be mounted by a remote systems.
- Run `exportfs -a` after modifying `/etc/exports`. The `-a` option tells `exportfs` to send all information in `/etc/exports` to the kernel.

You control the directory's availability by specifying a netgroup or host name; otherwise, the directory becomes available to everyone on the network running NFS. After accessing `/etc/exports`, the system checks `/etc/netgroup` for the netgroup definition; if it is not present, the system checks `/etc/hosts` for the host name. (For more information, refer to the previous sections, 4. Edit `/etc/hosts` and 5. Edit `/etc/netgroup`.)

Note If importing a directory containing a user's home directory, the user may not be able to login if the remote directory is not accessible.

If a client has a directory mounted and you edit `/etc/exports` and run `exportfs` to change availability of that directory, the client's access will not change. To prevent the client from accessing the server's files, on the client you must either unmount the directories or files or reboot the client.

`/etc/exports` contains a list of directories or files and the netgroup or machine names allowed to remotely mount each directory. The names are searched for in `/etc/netgroup` and then in `/etc/hosts`. A directory name with no accompanying name list means the directory is available to everyone.

<code>/etc/exports</code> Entry Formats	System Response
<code>/complete_directory</code>	Exports the directory or file to everyone on the network and defaults to <i>synchronous</i> writes on the NFS server.
<code>/complete_directory -access = client_1:client_2</code>	Exports to only these clients.
<code>/complete_directory -access = netgroup_1:netgroup_2</code>	Exports the directory or file only to the specified netgroups.
<code>/complete_directory -root = client_1:client_2</code>	Gives root access to only these clients.
<code>/complete_directory -anon = 0</code>	Gives all machines root access.
<code>/complete_directory -rw = netgroup_1:client_2</code>	Exports read-write access to the specified netgroup and client.
<code>/complete_directory -ro</code>	Exports read-only access to everyone on the network.

/etc/exports Entry Formats	System Response
<i>/complete_directory -access = client_1:client_2</i>	Exports the directory or file only to the specified clients.
<i>/complete_directory -access = client_2: netgroup_1</i>	Exports the directory or file only to the specified client and netgroup.
<i>/complete_directory -access = client_1,-async</i>	Exports the directory or file to the specified client and causes <i>asynchronous</i> writes on the NFS server.

EXAMPLE:

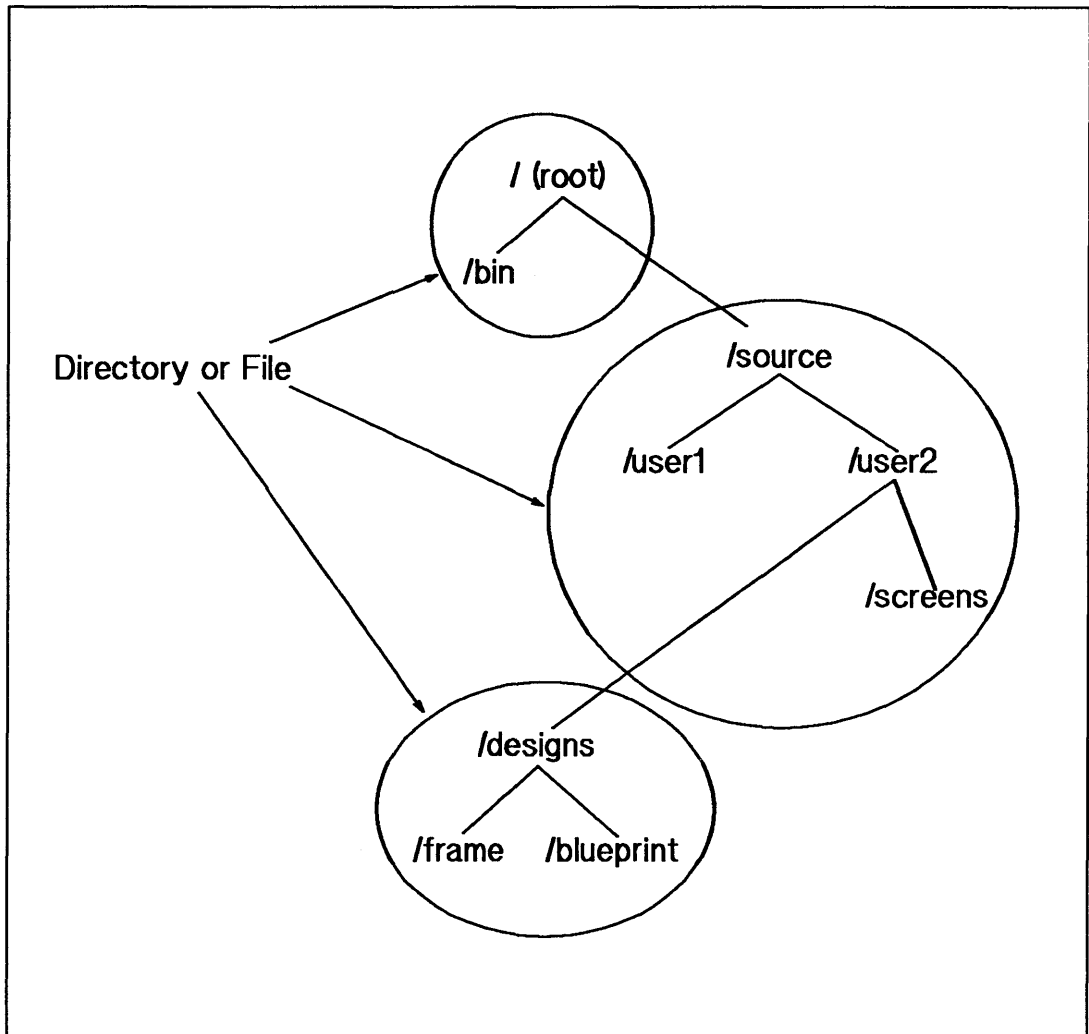


Figure 4-2. /etc/exports Example

/etc/exports Example Entry *	Example System Response
/	<p>Export / (root) to all clients.</p> <p>Clients will not receive the directory source since / is the exported directory. The source and designs directories or files will not be seen by this mount.</p> <p>Clients will receive the /bin directory since it is part of the / directory.</p>
/source	<p>Export source to all clients.</p> <p>Clients will not receive the directory designs or / since source is the exported directory. The designs and / directories or files will not be seen by this mount.</p>
/source/user2/designs	Export designs to all clients.
/source/user2/designs -async system1	Export designs to the client system1 and allow asynchronous writes on the NFS server.
/source/user2/designs lab	Export designs to the netgroup lab.
/source/user2/designs system1 lab	Export designs to the client system1 and the netgroup lab.
<p><i>*Note:</i> You must define all hosts in /etc/hosts and all netgroups in /etc/netgroup or, if you are using NIS, ensure that all hosts and netgroups are defined on the NIS master server.</p>	

NFS Export Options

Export options are preceded with a dash and are separated by commas:

```
/usr/new -access=homer:marge:bart,ro # export read-only to only these machines
```

Export options	Definitions
ro	export the directory read-only. This prevents hosts from writing to the filesystem.
rw = hostname:hostname	export the directory read-mostly. This limits read-write capability to only specified hosts. If the hosts are not specified, the directory is exported read-write to all. Up to 256 hostnames can be specified.
anon = uid	if a request comes from an unknown user, use uid as the effective user ID. Note: root users (uid 0) are always treated as "unknown" by the NFS server unless they are included in the root option below.
root = hostname[:hostname]	give root (superuser) access only to root users from a specified hostname or hosts. The default is for no hosts to be granted root access. Up to 256 hostnames can be specified.
access = client[:client]	allow mount access to the specified client or clients. A client can either be a hostname or a netgroup. Each client in the list is first checked in the netgroup database.
async	specifying async increases write performance on the NFS server by causing asynchronous writes on the NFS server. The async option can be specified anywhere on the command line after the directory name. Before using this option refer to the caution below.
#	a # character anywhere in the file indicates a comment that extends to the end of the line.

Caution The `-async` option increases write performance on the NFS server by allowing asynchronous writes on the NFS server's directory.

However, use caution in deciding whether to use the `-async` option. An unreported data loss may occur if the option is set and the NFS server hardware experiences a power loss, system panic or other failure.

Do not use the `-async` option with directories or files that contain:

- Files that are accessed by the `O_SYNCIO` flag (which is set by the `fcntl` or `open` calls).
- Data that cannot be reconstructed (e.g., a directory containing database files).
- Files synchronized with `fsync`.
- Critical applications requiring absolute data integrity.

if you are unsure whether any of the previous conditions apply, do not use the `-async` option.

Directory Exports

`exportfs` provides the ability to export a directory or single file for mounting over the network. It is invoked at boot time by the `/etc/netnfsrc` script, and uses information contained in the `/etc/exports` file to export a full directory. `/etc/exports` is read automatically by `exportfs`. If you modify the `/etc/exports` file you must rerun `exportfs -a` for the changes to take effect. `exportfs -a` can be run at any time to alter the list or characteristics of exported directories and files. Directories and files that are currently exported are listed in the file `/etc/xtab`.

The `exportfs` utility can also be used to withdraw an exported directory or filename. Stopping NFS access to a directory and filenames can be done by changing the `/etc/exports` file and executing `exportfs -a`. The changes will not take effect until you reboot or unmounting of a client occurs.

`exportfs` has the following command line options:

<code>-a</code>	exports all directories with their access permissions in <code>/etc/exports</code> .
<code>-u</code>	unexports the indicated directories. For example: <code>exportfs -u /source</code> denies NFS access to the <code>/source</code> directory.
<code>-o options</code>	executes a list of options and their parameters following the <code>-o</code> option. The list of options are the same options used for the <code>/etc/exports</code> file.
<i>directory</i>	The full pathname of the directory that you want to export or unexport.

Note A directory and one of its subdirectories cannot both be exported.

Mount Information

`exportfs` when executed, updates the `/etc/xtab` file. The `/etc/xtab` file determines what directories can be exported to NFS clients. Its contents change everytime the `exportfs` utility runs. The `mountd` daemon processes the `/etc/xtab` file each time a mount request is received. During the boot process, `mountd` checks `/etc/xtab`:

```
if [-f /etc/exports ]; then
    exportfs -a
    nfsd 8 &                echo -n ' nfsd'
    rpc.mountd -n
fi
```

Warning **The `/etc/xtab` file must not be edited because `rpc.mountd` problems will occur.**

`mountd` also provides information about which clients have filesystems mounted. This information can be printed using the `showmount(1M)` command.

Caution If a client crashes while executing `showmount(1)`, the server will show the client still having a filesystem mounted. The client's entry is not removed from the `/etc/xtab` until the client performs an explicit `umount` of the filesystem.

Also, if a client mounts the same remote directory twice, only one entry will appear in `/etc/xtab`. Doing a `umount` of one of these directories will remove the single entry and `showmount(1M)` will no longer indicate that the remote directory is mounted.

When using diskless capabilities, only the cluster server's `mountd` can respond successfully to mount requests. Cluster clients' `rpc.mountd` process is only used to answer `showmount(1M)` requests.

7. Execute `/etc/netnfsrc`

After you finish the configuration procedure, execute `/etc/netnfsrc` or reboot the system to activate the daemons and servers.

The rebooting process does not unmount any of the server's directories or files that were remotely mounted by other network nodes. However, these nodes will not be able to access any of the server's files until the server is operating again.

Create an NFS Client Manually (Without SAM)

You must be superuser to create an NFS client.

To create an NFS client, complete the following steps:

1. Edit `/etc/netnfsrc`.
2. Mount directories or files

An NFS client can also be configured as any combination of an NFS server, NIS client, or NIS server. (An NIS server *must* also be configured as an NIS client.)

1. Edit `/etc/netnfsrc`

The `/etc/netnfsrc` file activates the NFS daemons and servers. Do the following:

- To define the node as an NFS client, set the `NFS_CLIENT` variable to any digit other than zero.
- If the node is also a server, you may want to set the `NFS_SERVER` variable to any digit other than zero now. (Refer to the Create an NFS Server section to complete server configuration procedures).
- If the node is also a server for PC-NFS requests, set the `PCNFS_SERVER` variable to any digit other than zero.

Client Only	NFS_CLIENT = 1 NFS_SERVER = 0
Server Only	NFS_CLIENT = 0 NFS_SERVER = 1
Both Client and Server	NFS_CLIENT = 1 NFS_SERVER = 1
Neither Client nor Server	NFS_CLIENT = 0 NFS_SERVER = 0
PC-NFS Server	PCNFS_SERVER = 1

You can refer directly to the comments (lines beginning with # (pound) signs) for editing instructions and for descriptions of each activity executed by `/etc/netnfsrc`.

Note If you edit this file other than specified in this document, HP recommends you incorporate personal comments for future system administration.

```

#!/bin/sh
# netnfsrc      NFS startup file
###
# Depending on the configuration parameters you set within,
# this script sets up some or all of the following:
#* NIS specific:
# domainname    the NIS domain name
#
# and starts up some or all of the following programs:
# portmap      RPC (program_#,version) -> port_# mapper
# nfsd         NFS daemons
# biod         async BIO daemons
# pcnfsd       PC-NFS daemon
#* NIS specific:
# ypbind       NIS client process (all NIS nodes)
# ypserv       NIS server process (NIS server only)
# yppasswdd    NIS password daemon (NIS master server only)
###
# NFS_CLIENT    1 if this node is an NFS client, 0 if not
# NFS_SERVER    1 if this node is an NFS server, 0 if not
# Note: it is possible for one host to be a client, a server, both
# or neither! This system is an NFS client if you will be
# NFS mounting remote directories or files; this system is a server
# if you will be exporting directories or files to remote hosts.
# See Also: nfsd(1M), mount(1M)
###
# Note: this has nothing to do with whether or not the system is
# a rootserver or diskless client workstation. There is a
# test for this later.
###
NFS_CLIENT=0
NFS_SERVER=0
.
.
.
.
PCNFS_SERVER=0

```

2. Mount File Systems

Review the servers' `/etc/xtab` files on your LAN, FDDI, or Token Ring software to determine the directories or files to which you want the client to have access. You will need to mount each of these directories or files on the clients.

For each directory you should determine *one* of the following mounting methods:

- Mount automatically at boot time via `/etc/checklist`.
- Mount only when manually specified via the `mount` command.

Since an attempt to mount a remote directory requires using another node and the network, the mount may not succeed the first time. You can vary the number of times NFS attempts to mount a directory by using the `retry` option.

After the mount is successful, the manner in which NFS handles requests depends on whether the mount is hard (default) or soft.

NFS Hard Mount Hard mounting directories or files with the default `int` (interrupt) causes NFS to retry a request until it succeeds, you interrupt it, or you reboot the system. If the `noint` option is activated and an NFS server goes down, the system retries the request until the server comes up again or you reboot the system.

If the server does not respond to a hard mount request, NFS writes the following message in the network error log file.

NFS: server `host_name` not responding, still trying

Refer to following documentation for more error log information: *Installing and Administering LAN/9000*, *Installing and Administering Token Ring/9000*, and *Installing and Administering FDDI/9000*.

Note If a server that you previously performed a hard mount from goes down, you may not be able to access mounted directories or files on other nodes unless you reboot the problem server or interrupt all its requests.

NFS Soft Mount Soft mounting directories or files aborts requests after one attempt. NFS writes an error to the log file if the server does not respond to a request. The message varies depending on what type of request is made.

NFS server `host_name` not responding, giving up

NFS function `_name` failed for server `server_name`: TIMED OUT

Note When using a soft mounted directory, data can be damaged or lost when the directory is mounted read-write, even when using the retry count. To make NFS clients more accepting of soft mounted directories or files, try increasing the `retrans` mount option. Increasing the attempts to transmit the request to the server makes the client less likely to produce an RPC error during periods of server loading. To really guarantee data integrity, all directories or files mounted read-write should be hard-mounted.

If a user's home directory is in a remote directory, the user will not be able to login if the remote directory is not accessible (e.g., the server goes down, the network fails).

Mount Guidelines

Refer to the following guidelines whether mounting directories or files automatically via the `/etc/checklist` file or manually via the `mount` command. For more specific information, refer to `checklist(4)` and `mount(1M)` in the HP-UX Reference.

- You cannot mount a remote directory unless the server has an entry for your node in `/etc/exports` and `exportfs` has been run. (Execute `showmount` to list mounted directories or files.)
- A server can export directories.
- When you mount a new directory on top of a directory already containing files, the directory's files will no longer be accessible unless you execute `umount` to unmount the mounted directory.

To avoid masking a directory, HP recommends you mount the directory on top of an empty directory.

- You cannot unmount an open directory (a directory in which someone is currently operating).
- You must specify a mount point (name of a local directory on which the directory or file will be mounted).
- If operating in an HP-UX cluster environment:
 - If a cnode mounts a remote directory, all cnodes in the cluster can access the remote directory.
 - If using NFS to mount a directory attached to a cluster, you must use the host name where the directory is locally mounted as the node name specified in the `mount` command.

- If a cnode mounts a remote directory, any cnode in that cluster can unmount the remote directory.
- All mount points must exist on the directories or files mounted on the cluster root server. That is, mount points cannot exist on directories or files locally mounted on a cluster auxiliary server.
- If a cnode that mounted a remote directory goes down, all other cnodes in the cluster can still access that remote directory.
- Before mounting a directory, refer to the following table and determine the options you want the mount to have.
 - You must specify an option if mounting via `/etc/checklist`; you do not have to specify an option if mounting via `mount`.
 - You do not have to list options in a specific order; however, you must separate the options with commas (not spaces).

NFS Mount Options	Description
acdirmax = n	holds cached attributes for no more than <i>n</i> seconds after a directory update.
acdirmin = n	holds cached attributes for at least <i>n</i> seconds after a directory update.
acregmax = n	holds cached attributes for no more than <i>n</i> seconds after file modifications
acregmin = n	holds cached attributes for at least <i>n</i> seconds after file modifications.
actimeo = n	sets minimum and maximum times for regular files and directories to <i>n</i> seconds.
bg	<i>Background:</i> If the first request to a remote node's mountd fails, the mount process continues retrying the request in the background.
defaults	<i>Defaults:</i> The mount takes all the default options without you having to individually specify them. The defaults are noted within this table by asterisks (*). You only need to specify defaults when mounting via /etc/checklist; the mount command automatically provides the defaults.
devs*	<i>Devices:</i> Allows access to local devices.
fg*	<i>Foreground:</i> If the first request to a remote node's mountd fails, the mountd daemon retries the requests in the foreground.
hard*	<i>Hard Mount:</i> NFS retries until the request succeeds or you reboot the system. If you are using the intr default option, you can interrupt the directory request.
intr*	<i>Interruptable Mount:</i> You can press an interrupt key to abort an NFS request. (Though the interrupt key is not defined, common ones include [CTRL]-[C] and [BREAK].)
noac	Suppresses fresh client attributes when opening a file.
noauto	<i>No Automatic Mount:</i> Prevents the directory from being mounted when the mount -a option is executed. You only need to specify noauto when mounting via /etc/checklist.
nocto	Suppresses client attributes and name (look-up) caching.
nointr	<i>No Interruptable Mount:</i> You cannot interrupt processes waiting for NFS requests to complete.

NFS Mount Options	Description
nosuid	<i>No setuid:</i> You cannot execute files on the remote directory with either the setuid or setgid bits set.
port = <i>n</i>	Port = <i>n</i> Default <i>n</i> = 2049 (the NFS server port) Specifies the UDP port at which the NFS server is contacted. You should not have to reset this value.
quota	<i>Disk quotas:</i> Activate disk quotas on the directory.
* = <i>Default</i>	
retrans = <i>n</i>	Retransmit = <i>n</i> Default <i>n</i> = 4 When NFS sends a request to a remote system, RPC attempts to transmit the request <i>n</i> times. If RPC does not receive a response after <i>n</i> attempts, soft mounts return an error and hard mounts retry the request.
retry = <i>n</i>	Retry = <i>n</i> Default <i>n</i> = 1 The mount command retries mounting the directory <i>n</i> times; the default is 1. For example, if a mount attempt fails once and the default is 1, mount tries once more before quitting.
ro	<i>Read Only:</i> Access rights are <i>Read Only</i> .
rsize = <i>n</i>	Read requests size = <i>n</i> Default <i>n</i> = 8192 (8K) Specifies the maximum read request size used in communicating with the server.
rw*	<i>Read/Write:</i> Access rights are <i>read</i> and <i>write</i> .
soft	<i>Soft Mount:</i> NFS aborts the request after RPC attempts to transmit the request <i>n</i> times (as specified by the retrans option).

NFS Mount Options	Description
suid*	<p><i>setuid</i>: You can execute programs on the remote directory that have <i>setuid</i> as one of their permissions.</p>
timeo = <i>n</i>	<p>Timeout = <i>n</i></p> <p>Default <i>n</i> = 7</p> <p>Specifies the initial timeout (in tenths of seconds) for NFS requests.</p> <p>When an NFS request occurs, RPC sends the request, waits 0.7 seconds for a response, and then retries the request.</p> <p>After the initial timeout, the timeout increases by multiples of two each time no response is received. When a specified number of <i>retrans</i> retransmissions have been sent with no reply, soft mounts return an error and hard mounts retry the request.</p> <p><i>Note</i>: If performing NFS mounts through a gateway and you see several server not responding messages within a few minutes, change the timeout default value (7) to a value of 10 or greater until you stop seeing the message.</p>
wsize = <i>n</i>	<p>Write size = <i>n</i></p> <p>Default <i>n</i> = 8192 bytes (8K)</p> <p>Specifies the maximum write request size used in communicating with the server.</p>
* = <i>Default</i>	

Edit /etc/checklist for Mounts

If you want the directory mounted at system startup, add an entry for it in the /etc/checklist file. At boot time, /etc/netnfsrc2 executes `mount -at nfs` to mount all NFS directories or files listed in /etc/checklist.

Edit /etc/checklist to append the hosts and directories or files you wish to import using the following format. All of the default options are activated when you specify defaults. You must specify either defaults or at least one option.

For NFS Hard Mounts via /etc/checklist:

```
server_name:/imported_filesystem /mount_point nfs defaults 0 0
```

or

```
server_name:/imported_filesystem /mount_point nfs [options] 0 0
```

For NFS Soft Mounts via /etc/checklist:

```
server_name:/imported_filesystem /mount_point nfs soft[,other options] 0 0
```

The `nfs` stands for NFS mounts. NFS ignores the two zeros (0 0), though they must be present.

EXAMPLES: /etc/checklist Automatic Mounts

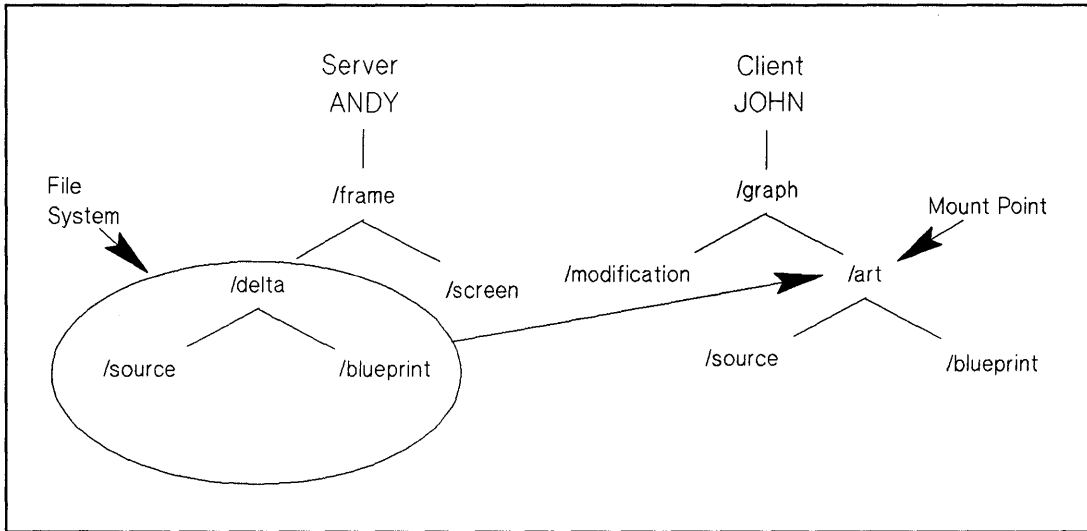


Figure 4-3. /etc/checklist Automatic Mounts

/etc/checklist Example Entry on the Client JOHN	Resulting Mount Options*
ANDY:/frame/delta /graph/art nfs defaults 0 0	Foreground Hard Mount Interruptable Port = 2049 Read and Write Read Size = 8192 Retransmit = 4 Retry = 1 setuid Timeout = 0.7 Write Size = 8192 <i>Note: All of these options are by default.</i>
ANDY:/frame/delta /graph/art nfs ro,retry=6,timeo=3 0 0	Read Only Retry = 6 Timeout = 0.3
ANDY:/frame/delta /graph/art nfs bg,retrans=8,soft 0 0	Background Retransmit = 8 Soft Mount
ANDY:/frame/delta /graph/art nfs noauto,noint,nosuid 0 0	No Automatic Mount No Interruptable Mount No setuid
ANDY:/frame/delta /graph/art nfs rsize=1024,wsiz=1024 0 0	Read Size = 1024 bytes Write Size = 1024 bytes
<p>* The default options are activated when you specify defaults. They are also active with other options unless you specify otherwise. The default options are listed only once for this example.</p>	

Execute mount for Manual Mounts

Execute `mount` to mount an NFS directory manually. NFS directories or files mounted via `mount` are only mounted as long as the client is running or until they are unmounted via `umount`. If the client goes down, you will have to re-mount the directory.

Do not use `mount` if you listed the directory in `/etc/checklist` since it will have already been mounted.

Use the following `mount` format for NFS mounts. All of the default options are activated unless you specify otherwise.

For NFS Hard Mounts via `mount`:

```
mount [-o options] server_name:/filesystem/mount_point
```

For NFS Soft Mounts via `mount`:

```
mount -o soft[,other_options] server_name:/filesystem/mount_point
```

EXAMPLES: `mount` Manual NFS Mounts

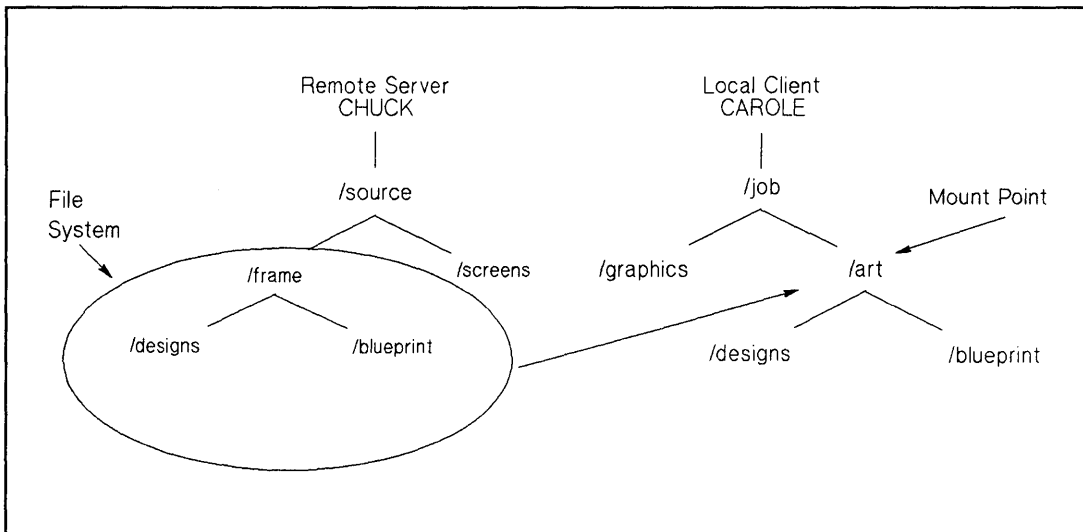


Figure 4-4. Example of a Manual NFS Mount

mount Example Command	Resulting Mount Options*
mount CHUCK:/source/frame /job/art	Foreground Hard Mount Interruptable Port = 2049 Read and Write Read Size = 8192 Retransmit = 4 Retry = 1 setuid Timeout = 0.7 Write Size = 8192 <i>Note: All of these options are by default.</i>
mount -o ro,retrans = 8,timeo = 3 CHUCK:/source/frame /job/art	Read Only Retransmit = 8 Timeout = 0.3
mount -o bg,retry = 6,rw,soft CHUCK:/source/frame /job/art	Background Read and Write Retry = 6 Soft Mount
mount -o noauto,noint,nosuid CHUCK:/source/frame /job/art	No Automatic Mount No Interruptable Mount No setuid
mount -o rsize = 1024,wsiz e = 1024 CHUCK:/source/frame /job/art	Read Size = 1024 bytes Write Size = 1024 bytes
* All of the default options are activated unless you specify otherwise. The default options are listed only once for this example.	

3. Execute `/etc/netnfsrc`

After you finish the configuration procedures, execute `/etc/netnfsrc` or reboot the servers and clients to activate the daemons and servers.

The rebooting process unmounts all local directories or files and directories that were manually mounted by the client (i.e., were not automatically mounted by `/etc/checklist`).

Configure NIS (optional)

If you plan to use the optional Network Information Service (NIS), refer to the NIS Configuration and Maintenance chapter for detailed configuration procedures.

Configure VHE (optional)

If you plan to use the optional Virtual Home Environment (VHE) service, refer to the VHE Configuration and Maintenance chapter for detailed configuration procedures.

Execute `/etc/netnfsrc`

To complete the configuration procedure, execute `/etc/netnfsrc` (or reboot) your system.

Note You have completed configuring the base NFS service. Refer to the remaining part of the chapter for maintenance information.

NFS Maintenance

To keep NFS running correctly and efficiently, refer to the following sections to ensure it stays configured to meet your changing needs:

- Maintain NFS Services using SAM.
- Prevent systems from accessing local directories or files via NFS (without using SAM).
- Update software.
- Clock skew.
- Maintain the NFS server.

Maintain NFS Services Using SAM

SAM (System Administration Manager) provides an automatic method for maintaining your local system's client and server services. You must be super-user to use SAM.

All NFS configurations available in SAM are done in the *Networked File Systems* window. This section explains how to move to the NFS window where you can select the task you wish to perform.

1. At the HP-UX prompt, type: `sam`

Wait for SAM's main menu to appear.

To Modify or Remove Connectivity Information about a Remote System:

1. At the SAM Main window, highlight *System-to-System Connectivity* and activate the **OPEN** button.

This window allows you to modify connectivity information about a remote system. You will be able to modify entries from `/etc/hosts` and possibly modify `/etc/route` entries from `/etc/netlinkrc`, depending on the information you are modifying.

Note the following information before you begin:

- If your system is configured to use the NFS Network Information Service or the ARPA Service's BIND Name Service for hostname-to-address mapping, you cannot use SAM to modify NFS Services connectivity information about a remote system. This window edits only the `/etc/hosts` file; it does not edit an NFS Network Information Service or BIND Name Service database.

- If you were using a specified gateway, other than the default gateway, to reach the remote system you are modifying connectivity information about, SAM will automatically modify or remove the special routing information by executing an `/etc/route delete host` command and modify or remove it from `/etc/netlinkrc`.
1. Highlight Internet Connectivity and activate the **OPEN** button. This window will display the remote system names and IP address that are already configured.
 2. From the object list displayed, highlight the remote system name you wish to modify or remove, choose either **modify** or **remove** from the "Actions" menu and then enter the information about the system you want to modify or remove connectivity with.
 3. When you are finished entering the remote system information, activate the **OK** button. This will return you to the previous window.
 4. Activate the **OK** button to return to the System-to-System Connectivity window. SAM updates the list of remote system to include the information you configured.
 5. To return to the Network/Communications window, select and activate the **Previous Level** button.

Prevent systems from accessing local directories or files via NFS using SAM (Stop Being an NFS Client)

This allows you to set up your local system so that it is no longer an NFS client. When you perform this task you are editing the `/etc/netnfsrc` file killing all `/etc/biod` daemons, and unmounting all NFS-mounted file systems.

1. From the Networking/Communications window, highlight *Networked File Systems (NFS)* and activate the **OPEN** button.
2. Highlight *Remote File Systems Mounted* and activate the **OPEN** button.
3. To modify, remove or disable your NFS client, choose either **Modify**, **Remove**, or **Disable NFS Client** from the "Actions" menu and then enter or remove the information about the remote directory you are mounted to.
4. When you are finished modifying or removing information, activate the **OK** button to return to Remote File Systems Mounted window.
5. If you choose the **Disable NFS Client** from the "Actions" menu, you will see under the title, Remote File Systems Mounted, NFS Client: Disabled, displayed above the object list.

Prevent remote systems from accessing local directories or files via SAM (Stop being an NFS Server)

1. From the Networking/Communications window, highlight *Networked File Systems (NFS)* and activate the **OPEN** button.
2. Highlight *Local File Systems Exported* and activate the **OPEN** button.
3. To modify or disable your NFS server, choose either **Modify** or **Disable NFS Server** from the "Actions" menu and then enter or remove the information about the local directory you are exporting.
4. When you are finished modifying or removing information, activate the **OK** button to return to Local File Systems Exported window.
5. If you choose the **Disable NFS Server** from the "Actions" menu, you will see under the title, Local File Systems Exported, NFS Server: Disabled, displayed above the object list.

Prevent systems from accessing local directories or files via NFS (without using SAM)

You may need to prevent file access via NFS from either one of the following situations:

- The client to keep local users from accessing NFS-mounted remote directories or files.
- The server to keep all clients from accessing local directories or files via NFS.

Unmount File Systems from Client

If needed, you can unmount directories or files on a client. Unmounting directories or files prevents further access to the server's files until you remount the directory. Read the following:

- Executing the `umount` command unmounts directories or files mounted either via `mount` or `/etc/checklist`.
- You cannot unmount an open directory or a parent of an open directory (e.g., a directory in which someone is currently operating).
- If operating in an HP-UX cluster environment:
 - If a cnode mounts a remote directory, any cnode in that cluster can unmount the remote directory.
 - If a cnode unmounts a directory, all cnodes in the cluster will have that directory unmounted.

Unmount File Systems on Clients	Action
One NFS directory on a client.	On the client, execute <code>umount</code> : <code>umount <i>mount_point_name</i></code>
All directories or files on one client.	On the client, execute <code>umount -a</code> : <code>umount -a</code> <i>Note:</i> This command unmounts all directories or files, not just NFS directories or files. If operating in an HP-UX cluster environment, clients should not execute <code>umount -a</code> .
All NFS directories or files on all clients	On all clients, execute <code>umount -at</code> : <code>umount -at nfs</code>
All directories or files listed in <code>/etc/mnttab</code> that were remotely mounted from a specified server	On all clients, execute <code>umount -h</code> . <code>umount -h <i>server_name</i></code>

Prevent Access to Server File Systems

If needed, you can prevent clients from accessing directories or files on the network servers.

Prevent Access to Server Directories or Files	Action
One NFS directory from a client	<ol style="list-style-type: none"><li data-bbox="705 348 1168 696">1. You have two options for Step 1:<ul style="list-style-type: none"><li data-bbox="705 413 1168 534">- If a netgroup is specified for that file system in <code>/etc/exports</code>, remove the host name from the netgroup entry in the server's <code>/etc/netgroup</code> file.<li data-bbox="705 574 1168 696">- If a host name is specified for that file system in <code>/etc/exports</code>, remove the host name from the server's <code>/etc/exports</code> file. Then run the <code>exportfs</code> command.<li data-bbox="705 736 1076 822">2. On the client, execute <code>umount</code>. <code>umount mount_point_name</code> <p data-bbox="705 864 1229 951"><i>Note:</i> A directory can not be unmounted if a client has opened it. Access will be denied the next time the client accesses the directory.</p>
One NFS directory from a netgroup	<ol style="list-style-type: none"><li data-bbox="705 986 1200 1107">1. On the server, remove the netgroup name (associated with that directory) from either the <code>/etc/exports</code> file or from <code>/etc/netgroup</code>.<li data-bbox="705 1147 1096 1173">2. On all members in the netgroup.

Prevent Access to Server Directories or Files	Action
All NFS directories or files from all clients	<p>1. On all clients, execute <code>umount</code>.</p> <p style="padding-left: 40px;"><code>umount mount_point_name</code></p> <p>2. On the server, you have two options for Step 2:</p> <ul style="list-style-type: none"> - Kill the <code>nfsd(1M)</code> daemon or daemons (usually four); the system prohibits NFS accesses only until you restart the <code>nfsd</code> daemons or you reboot the system. - Edit <code>/etc/netnfsrc</code> to change the <code>NFS_SERVER = value</code> to zero, and reboot the system. <p><code>NFS_SERVER = 0</code> or <code>exportfs -ua</code> (unexports all directories or files).</p>

Update Software

To install a new system release to a server, use the `/etc/update` program to install software. (Refer to the *HP-UX System Administrator's Manual* for detailed instructions.)

The following list includes configuration files loaded during the `/etc/update` process. Some of these files contain example entries to help you configure them correctly:

- `/etc/checklist`
- `/etc/netnfsrc`
- `/etc/inetd.conf`

- /etc/rpc
- /etc/netgroup
- /usr/adm/inetd.sec
- /etc/netnfsrc2

Note If you are mounting directories or files, then load *only* those file sets that reside on the local directories or files.

For 8.0, /etc/netnfsrc has changed significantly. A new /etc/netnfsrc file is loaded during the install and update procedure. Standard values are propagated from the old version to the new version. The old version is saved in /etc/netnfsrc.OLD. You must copy any customization necessary from the old version.

When using /etc/update, the system creates new configuration files in the /etc/newconfig directory. These files correspond to the original configuration files which the system leaves in /etc.

- Compare each file in /etc/newconfig with its existing counterpart in /etc to determine if you need to update or replace the file.
- If needed, edit the /etc/newconfig files to meet your specific needs.
- Once the /etc/newconfig file suits your configuration needs, replace the existing file in /etc with the new one in /etc/newconfig.
- You may want to save the old configuration file for later reference.

Clock Skew

The NFS client and server clocks may not be synchronized since each workstation keeps its own time. Problems may occur because of these time differences.

If your application depends on the local time or directory timestamps, then it may have to handle clock skew problems if it uses remote files. For example, when giving `utime` a NULL pointer for the times value, the following process occurs:

1. The system sets the access time and modification time according to the client node clock.
2. It then sends these times over to the server, which then changes the inode to reflect the new access and modification times.
3. The server node identifies the change in the inode and thus, modifies the inode's status change time according to its own clock.

The result is a high probability of differing times between the file or directory's access and modification times versus its status change time.

Note HP corrected the clock skew problems that existed with the `ls` command and the source code control command `SCCS`.

If operating in an HP-UX cluster environment, all nodes in the cluster have the same time as the root server's clock. Therefore, clock skew problems exist only if the root server's clock is different from other NFS servers.

EXAMPLE: This example shows how a command could be affected by the clock skew.

Problem Most programs logically assume an existing file could not be created in the future; one example is `ls`. (Note: This example shows how HP corrected this problem.)

The `ls -l` has two basic forms of output, depending on how old the file is.

```
$ date
April 7 15:27:31 PST 1987
```

```
$ ls -l file*
-rw-r--r-- 1 root other Aug 26 1981 file (Form One)
-rw-r--r-- 1 root other Apr 07 15:26 file2 (Form Two)
```

Form One of `ls` prints the month, day, and year of the last file modification if the file is *more* than six months old. Form Two prints the month, day, hour, and minute of the last file modification if the file is *less* than six months old.

The `ls` command calculates the age of a file by subtracting the modification time of the file from the current time. If the results are greater than six months, the file is old.

Now assume that the time on the server is three minutes ahead of the local node's time (April 7, 15:30:31). The following commands demonstrate the effect of this clock skew prior to HP's correction of the problem.

```
$ date
April 7 15:27:31 PST 1987
$ touch file3
$ ls -l file*
-rw-r--r-- 1 root other 0 Aug 26 1981 file
-rw-r--r-- 1 root other 0 Apr 07 15:26 file2
-rw-r--r-- 1 root other 0 Apr 07 1987 file3
```

The problem is that the difference of the two times is negative, but the variable in the computation is unsigned. A signed negative number has the same representation (bit pattern) as a very large unsigned number.

local node time = 15:27:31
modification time = local node time plus 180 seconds

local node time	15:27:31
- modification time	- (15:27:31 + 180)

large unsigned number that	- 180 seconds
is greater than six months	

Problem
Correction

HP corrected the problem so that ls now prints the month, day, and minute for files between six months old and one hour ahead of time. Other applications may also require such modification.

```
$ date
April 07 15:27:31 PST 1987
$ touch file3
$ ls -l file*
-rw-r--r-- 1 root other 0 Aug 26 1981 file
-rw-r--r-- 1 root other 0 Apr 07 15:26 file2
-rw-r--r-- 1 root other 0 Apr 07 15:30 file3
```

Maintain the NFS Server

NFS servers are described as stateless. This means the NFS server does not know and does not care which clients import its directories or files. This gives NFS the advantage of allowing clients to access NFS mounted directories or files after the server has recovered from a crash.

To recover as a client, all you need to do is try an NFS remote procedure call by accessing an NFS mounted file.

To recover as a server, just reboot the system (assuming that you have networking and `nfsd` configured to start at boot time).

However, problems occur when servers remain down for extended periods of time and/or directories or files are rearranged or modified while the servers are down. In these cases, clients can hang on NFS mounts. In extreme cases, servers' directories or files may be corrupted when they become available to NFS clients. There are some precautions you can take to reduce the problems experienced during server maintenance and server crashes. These precautions are covered in the following sections.

Planned Downtime

Before bringing down an NFS server, HP recommends that you unmount all NFS clients. This ensures that no clients hang on a server that is not responding. It also protects the server from possible directory corruption if directories or files are modified while the server is down.

However, unmounting NFS clients can be a problem. Unless you restrict access to a small group of clients, it is difficult to determine which clients have directories or files imported from a particular server. The `/etc/rmtab` file on the server will give some clues which nodes might be clients, but it is not reliable. The only way to get a complete list of clients is to do an exhaustive search of all possible clients. Do this by logging onto all possible clients, executing the `mount` command, and searching the output for evidence of a directory imported from the server.

If directories or files are modified or rearranged on the server while the server capabilities are disabled and not all clients have unmounted the server's directories or files, directory corruption may occur when the server resumes servicing NFS requests. To reduce the possibility of directory corruption, run `fsirand` on the server's directories or files while they are unmounted locally. (See `fsirand(1M)` in the *HP-UX Reference*.) This will randomize the inode generation numbers on the directories or files thereby minimize the possibility of NFS clients incorrectly modifying these directories or files after NFS Services resume on the server. This will cause clients which have the files mounted to see "Stale file handle" messages. These clients must unmount and possibly reboot to continue accessing the NFS mounted directory.

Unplanned Downtime

When NFS servers crash unexpectedly, NFS clients obviously cannot access directories or files imported from the crashed servers. If the clients have hard-mounted NFS directories or files, which is the default, client applications that attempt to access those directories or files will hang. After rebooting the server, the applications will continue as normal.

Hanging applications can be an annoyance if servers are down for a short period of time. However, hanging applications can become a big problem if servers are down for long periods of time. For this reason, the soft mount option is provided (see "Create an NFS Client Manually"). The soft mount option allows you, as the NFS client system administrator, to determine how many times an NFS request should be transmitted before giving up. This allows applications to continue processing when access to NFS directories or files is impossible due to a crashed server.

When NFS servers are not brought down gracefully, there is always the possibility of directory corruption. If the damage is significant, you may need to recreate the directory. If this is the case for a directory that is exported via NFS, it is important to run `fsirand` on the directory after it is recreated and before it is mounted locally. This will add a level of protection on the directory when the NFS client applications attempt to access the directory.

Note Running `fsirand` does not guarantee that a directory corruption will not occur when NFS clients attempt to access the directory. However, it does significantly reduce the possibility.

Remote Execution Facility (REX)

This chapter describes how to configure and execute commands on a remote host using the Remote Execution Facility (REX).

REX consists of:

- The `on` command
- The `rex`d (remote execution daemon)

The `on` command provides the REX user interface on the client. It also communicates with `rex`d to execute commands remotely. `rex`d runs on the server and facilitates the execution of the remote commands.

The functionality of REX is similar to that of remote shell (`remsh`) with two important differences:

- REX executes commands in an environment similar to that of the invoking user. Your environment is simulated by:
 - Copying all of your environment variables to the remote computer.
 - Mounting the directory containing your current working directory on the remote computer via NFS (if it is not already mounted on the remote computer). Your command is then executed on the remote computer in the remote version of your working directory, using your (the invoking user's) environment variables.
- REX allows you to execute interactive commands such as `vi`. In this case, your current `tty` settings (e.g. your current break character) are also copied to the remote system.

The on Command

The `on` command provides the user interface for remote execution of commands. When executing the `on` command, you specify:

- A host on which to run the remote command.
- The command to run.
- Arguments for the command.

The `on` command then simulates your current environment on the server by passing your environment variables and information about your current working directory to the remote host. The `rexd` daemon on the server mounts the directory that contains your current working directory if it is not already mounted on the server. After the environment is simulated, the command executes in the simulated environment on the remote host.

When a remote system is `nfs-mounted`, `rexd` sets up a default path, a default shell which is a subset of the local shell environment rather than the full customized shell environment. The default path consists of the remote current directory that is mounted to the remote system, `/usr/spool/rexd/reXXXXX/currentdir`. `rexd` assumes the remote directory is in the same path as the local host when line commands like `cd` or `alias` are used. If this is not the case, `rexd` returns an error, command not found and the command will be executed relative to the current local path.

It is a good idea to do a `cd <remote dir>` to establish a reference point on the remote host before you try giving commands involving the directory. This will take you out of the remote `nfs rexd` mount point and into the remote system at a point where you can expect proper behavior of the commands.

Note Your environment is *simulated* on the remote host but not completely recreated. Execution of a given command on a remote host will not always produce the same results as executing the command on your local system because it is not always known where `rexd` will put the mount point and often the paths across machines are not the same. Unless you know how the local and remote hosts are configured you are better off not using any command that you know needs an absolute path or a relative path which depends on a local and remote configuration match. The simulated environment and the environment's limitations are discussed below in Environment Simulation.

The syntax of the `on` command is as follows:

```
on [-i | -n] [-d] host [command [argument] ....]
```

5-2 Remote Execution Facility (REX)

Host specifies the name of the host on which to execute *command*. There must be an entry for *host* in the local computer's host data base.

Command specifies the command to execute on *host*. If *command* is not specified, *on* will start a shell on *host*.

You may specify three options (*-i*, *-n*, *-d*). The *-i* option must be used when invoking interactive commands, the *-n* option must be used when running commands in the background with job control, and the *-d* option is used when you wish to receive diagnostic messages.

Use of the *-d* option with either *-i* or *-n* is permitted.

EXAMPLE:

```
on -i -d host
```

or

```
on -n -d host
```

You *cannot* use the *-i* and *-n* options at the same time.

The -i Option (Interactive Mode)

The *-i* option invokes the interactive mode. This option must be specified for all interactive commands (commands which expect to be communicating with a terminal). Examples of interactive commands are *vi*, *cs*, and more. If this option is specified with a non-interactive command such as *sort*, it will be executed as an interactive command, but there may be no difference in behavior.

EXAMPLE:

```
on -i node_7 vi <file>
```

The -n Option (No Input Mode)

The *-n* option sends the remote program an end-of-file when the program reads from standard input instead of connecting the standard input (*stdin*) of the *on* command to the standard input (*stdin*) of the remote command. The *-n* option is necessary when running commands in the background with job control.

The -d Option (Debug Mode)

The -d option allows you to receive diagnostic messages during the start up of the on command. The messages may be useful in detecting configuration problems if the on command is failing while connecting to a given host.

Configuration Requirements

The following list details the configuration requirements that must be met for you to execute the on command from node A to node B:

- You must be logged into a user account (other than root) on node A.
- You must have an account on node B, and the UIDs for the accounts on node A and node B must be the same. If this is not the case, one of two things will happen:
 - If the UID associated with the user on node A is not associated with any user on node B, the on command will fail with the error:

on hostname: rexd: User id xxxx is not valid.
 - If your UID on node A is associated with another user on node B, then the command will be executed on node B as the user associated with the UID. (The second case is a serious security limitation. More details are given in the Security Considerations section of this chapter).
- The directory that contains your current working directory must be exported in a manner that allows computer B to mount it. Note that the current working directory may be a directory on another remote computer C, which is being accessed via NFS.
- Node B must have rexd configured to execute.

Environment Simulation

As previously mentioned, your environment is simulated on the remote computer, not mirrored. Therefore, certain limitations exist that may cause the execution of a given command to produce different results or errors when executed on the local computer and a remote computer via `on`. These limitations are as follows:

- If the directory is not already mounted on the remote computer, the directory containing your current working directory will be mounted on the remote computer in a subdirectory of `/usr/spool/rexd`. If the directory is already mounted on the remote computer, the mount point is the current mount point for the directory. Therefore, the use of absolute path names can cause problems.

EXAMPLE:

User `mjk` on node `A` is in his home directory (`/users/mark/mjk`) and executes the `on` command to start a shell on a remote system. When the shell is started, the current directory will be `/usr/spool/rexd/rexdAXXXX/users/mark/mjk` (where `A` is a letter and `XXXX` is a 4 digit number). If `mjk` now types the command `cd`, one of two events will occur, depending on the configuration of the directory on the remote computer:

- If the path `/users/mark/mjk` exists on the remote system, the current directory will be `/users/mark/mjk` on the remote system, which is not equivalent to `/users/mark/mjk` on the local system.
- If the path `/users/mark/mjk` does not exist on the remote system then executing `cd` will return an error.

This type of behavior could cause a script that executes `cd` or uses absolute file names to produce different results when executed remotely.

- Another example where the use of absolute path names may occur, without being obvious, is the use of `$PATH`. Implicit use of `$PATH` may cause a different version of a command (or a different command) to be executed in the remote case.
- Relative path names will work if they are within the same directory as your current working directory. If a relative path name crosses a directory boundary it will encounter problems similar to those presented by use of absolute path names.

Configuring rexd

Configuring `rexd` on a system allows the system to act as a server, executing commands for clients that execute an `on` command. Before configuring `rexd` to run on a system, you should read the Security Considerations section in this chapter.

When `rexd` is configured, it is started by `inetd` when a request for remote execution is made by a client. `inetd` obtains the information it needs to start `rexd` from the file `/etc/inetd.conf`. The following entry must be in the file `/etc/inetd.conf` in order for `inetd` to start `rexd`:

```
rpc stream tcp nowait root path 100017 1 rpc.rexd [ options ]
```

Path and *options* are defined as:

- | | |
|----------------|---|
| <i>path</i> | The path name of the <code>rexd</code> executable in the directory. The <code>rexd</code> shipped with the HP NFS Services product is located in <code>/usr/etc/rpc.rexd</code> . |
| <i>options</i> | The options that change the behavior of <code>rexd</code> . Each of the possible options is described below: |

The `-l` option

You can log any errors reported by `rexd` to a file by adding `-l log_file` at the end of the configuration entry in `/etc/inetd.conf`, where `log_file` is the name of the file where errors are logged. If `log_file` exists, `rexd` appends messages to the file. If `log_file` does not exist, `rexd` creates it. Messages are not logged if the `-l` option is not specified.

The information logged to the file includes the date and time of the error, the host name, process ID and name of the function generating the error, and the error message. Note that different RPC services can share a single log file since enough information is included to uniquely identify each error.

EXAMPLE:

Thus, the entry in `/etc/inetd.conf` to log errors to the file `/usr/adm/rexd.log` is:

```
rpc stream tcp nowait root /usr/etc/rpc.rexd 100017 1 \  
rpc.rexd -l /usr/adm/rexd.log
```

The -m option

Specifying `-m mountpoint` changes the default directory containing mount points. This directory is used for mounting client directories or files. The following entry in `/etc/inetd.conf` causes client directories or files to be mounted as `/client/mnt/rexdAXXXX` instead of `/usr/spool/rexd/rexdAXXXX` (where *A* is a letter and *XXXX* is a 4 digit number):

```
rpc stream tcp nowait root /usr/etc/rpc.rexd 100017 1 rpc.rexd -m /client/mnt
```

The owner, group, and all other users must have read and execute permission for *mountpoint* or an `on` command may fail for a user that does not have the proper permission to *mountpoint*.

The -r option

The `-r` option causes the `rex`d to use stronger security checking than it uses by default (see Security Considerations). When started with the `-r` option, `rex`d denies access to a client unless one of the following conditions is met:

- The name of the client is in the `/etc/hosts.equiv` file on the server.
- The user on the server, associated with the UID sent by the client, has an entry in `$HOME/.rhosts` that specifies the client name followed by *one* of the following:
 - White space and an end of line.
 - or
 - The user's name and an end of line.

EXAMPLE:

If a user assigned to UID 7 on NODE1 executes the following on command:

on NODE2 pwd

Then user mjk (assuming user mjk on NODE2 is assigned UID 7) on NODE2 must have *one* of the following entries in \$HOME/.rhosts:

NODE1

or

NODE1 mjk

Security Considerations

The design and implementation of REX incorporates several security limitations that you should consider before configuring `rex`d. REX restricts access to a system by use of UIDs. That is, the client (`on`) passes the invoking user's UID to the server (`rex`d) to determine if the invoking user is a valid user. This creates several security limitations:

- If the client and the server do not have the same mapping of user to UIDs, a user on a client may be able to access the server as some other user.
- A malicious user can set the desired UID in the outgoing packets and access the server as any of the server's valid users other than root. An individual with their own workstation can set up a user account with the desired UID.

The impact on system security can be reduced by using the file `/usr/adm/inetd.sec`. The entries in this file specify a set of networks and hosts that are allowed or denied access to a service that is started by `inetd`. For more details on the use of `/usr/adm/inetd.sec` see *inetd.sec(4)* in the *HP-UX Reference*.

The consequences can also be reduced by use of the `-r` option when starting `rex`d. See the previous section, *Configuring rex*d, for more details about the `-r` option.

Under normal NFS use, only root is allowed to mount remote directories or files. However, when `rex`d is in use, you can mount a directory on the server by executing the following instructions:

1. `cd` to a directory in the directory you wish to mount.
2. Execute the `on` command to start a shell on the computer on which you wish to mount the directory.
3. From another window, shell layer, or system, log into the server and `cd` to a directory in the directory that `rex`d mounted.
4. Switch back to the previous window, shell layer, or system and exit the shell created by the `on` command.

Since another user is busy in the mounted directory, `rex`d will be unable to unmount the directory. Hence, the user has mounted the directory.

Diagnostics

There are two types of error messages discussed in this section. They are:

- on command error messages.
- rexd error messages.

on Command Error Messages

The following on command error messages are sent to `stderr`.

on: unknown host *<host>*

The host name *<host>* was not found in the *hosts* database.

on: cannot connect to server on *<host>*

The host *<host>* is down, unreachable on the network or not running rexd.

on: can't find *<current_dir>*

A problem occurred trying to find your current working directory (*<current_dir>*).

on: can't locate mount point for *<current_dir>*

A problem occurred trying to determine the mount point of your current working directory (*<current_dir>*).

on: standard input (stdin) is not a tty

The standard input (stdin) of the on command with the `-i` option is not a tty.

on *<server>* : rexd: *<message>*

Errors which occur on the server *<server>* are propagated back to the client. These messages are documented in the DIAGNOSTICS section of *rexd(1M)* found in the *HP-UX Reference*.

rexd Error Messages

The following is a subset of the messages that may appear in the log file if the `-l` option is used. Some of these messages are also returned to the client.

rexd: could not umount *<dir>*

rexd was unable to umount your current working directory. See *rexd(1M)* in the *HP-UX Reference* for more details.

rexd: mountdir (*<mountdir>*) is not a directory

The path name *<mountdir>*, under which temporary mount points are created, is not a directory or does not exist.

rexd: *<command>*: Command not found

rexd could not find *<command>*.

rexd: *<command>*: Permission denied

rexd was denied permission to execute *<command>*.

rexd: *<command>*: Text file busy

The executable file is currently open for writing.

rexd: *<command>*: Can't execute

rexd was unable to execute *<command>*.

rexd: root execution not allowed

Root execution is not allowed by rexd.

rexd: User id *<UID>* not valid

The UID *<UID>* is not assigned to a user on the server.

rexd: User id *<UID>* denied access

rexd was started with the `-r` option, and the remote execution request did not meet either of the conditions required by the `-r` option.

rexd: *<host>* is not running mountd

The host *<host>* on which the user's current working directory is located is not running mountd. Therefore, rexd is unable to mount the required directory.

rex: not in export list for <*directory*>

The host on which the client's current working directory is located does not have the server on the export list for the directory <*file_system*> containing the client's current working directory. Therefore, rex is unable to mount the required directory.

The Network Lock Manager

This chapter explains file and record locking using the Network Lock Manager (`rpc.lockd`) and the Network Status Monitor(`rpc.statd`). It also explains how file locking is used to synchronize access to shared files.

File and record locking allows cooperating processes to synchronize access to shared files. You interface with the networking service by way of the standard `lockf()` system call interface, and rarely require any detailed knowledge of how it works. The operating system maps user calls to `lockf()` and `fcntl()` into Remote Procedure Call (RPC)-based messages to the local lock manager. The fact that the directory may be located on a different node is not really a complication—until a failure occurs.

All computers fail or simply shut down from time-to-time, and in an NFS environment, where multiple computers can have access to the same file at the same time, the process of recovering from a failure is necessarily more complex than in a non-networked environment. Furthermore, locking is inherently stateful. If a server fails, clients with locked files must be able to recover their locks. If a client fails, the locks will be released when the client comes back up. To preserve the overall transparency of NFS, the recovery of lost locks must not require the intervention of the applications themselves. This is accomplished as follows:

- Basic file access operations, such as read and write, use a stateless protocol (the NFS protocol). All interactions between NFS servers and clients are atomic—the server doesn't remember anything about its clients from one interaction to the next. In the case of a server failure, client applications will sleep until the server recovers and NFS operations can complete.
- Stateful services (those that require the server to maintain client information from one transaction to the next) such as the locking service, are not part of NFS. They are separate services that use the status monitor (see The Network Status Monitor section at the end of this chapter) to ensure that their implicit network state information remains consistent with the real state of the network. There are two specific state-related problems involved in providing locking in a network context:
 - If the client has failed, the lock can be held forever by the server.
 - If the server has failed, it loses its state (including all its lock information) when it recovers.

The Network Lock Manager solves both of these problems by cooperating with the Network Status Monitor to ensure that it is notified of relevant computer failures. The Lock Manager

protocol then allows it to recover the lock information it needs when a computer recovers from a failure.

Structure of the Network Locking Service

The following illustration depicts the overall structure of the network locking service.

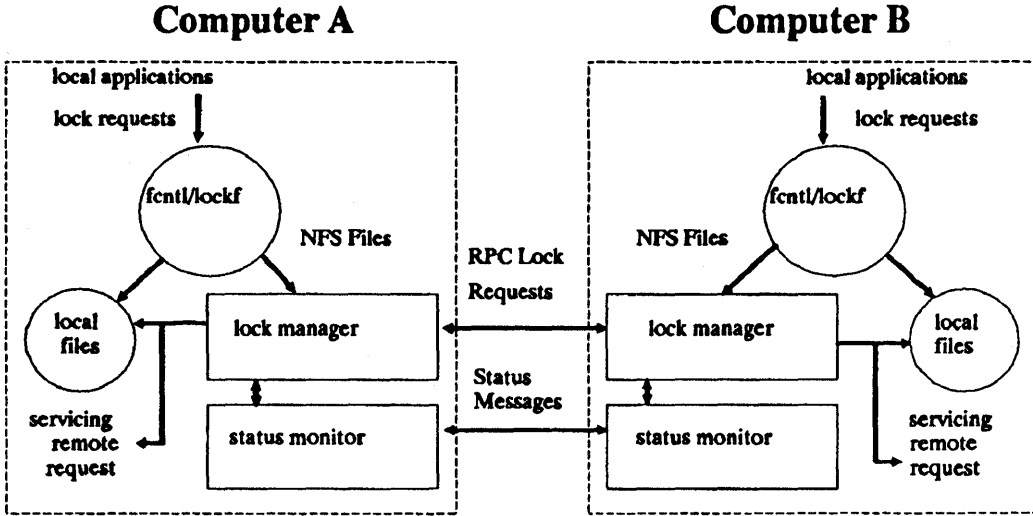


Figure 6-1. Architecture of the Network Locking Service

At each server site, a lock manager process accepts lock requests, made on behalf of client processes by a remote lock manager. The client and server lock managers communicate with RPC calls. When the lock manager receives a remote lock request for a computer that it doesn't hold a lock for, the lock manager registers interest in that computer with the local status monitor. The lock manager then waits for notification from the local status monitor that the computer is up. The local status monitor continues to watch the status of registered computers, and notifies the lock manager when one of them is rebooted (after a failure). If the lock request is for a local file, the lock manager tries to satisfy it, and communicates back to the application along the appropriate RPC path.

If the failure of a client is detected, the server releases the failed client's locks, on the assumption that the client application will request locks again as needed. If the recovery (and, by implication, the failure) of a server is detected, the client lock manager retransmits all lock requests previously granted by the recovered server. This retransmitted information is used by the server to reconstruct its locking state. See the Locking Protocol section for more detail.

The locking service, then, is essentially stateless. Or to be more precise, its state information is carefully circumscribed within a pair of system daemons that are set up for automatic application-transparent failure recovery. If a server fails, and thus loses its state, it expects that its clients will be notified of the failure and send it the information that it needs to reconstruct its state. The key in this approach is the status monitor, which the lock manager uses to detect both client and server failures.

Note Failure detection and recovery cannot occur until the system with the failure is rebooted.

Starting the Network Locking Services

The network locking service daemons, `rpc.lockd` and `rpc.statd` are started at boot time by the `/etc/netnfsrc` script. If `/etc/netnfsrc` is configured to make this node an NFS client and/or server, `rpc.lockd` and `rpc.statd` are started by default. If this is not the desired behavior, the `/etc/netnfsrc` script can be modified to produce the desired behavior.

It is important that `rpc.lockd` and `rpc.statd` are running on both the client and server for the file locking protocol to operate correctly.

The Locking Protocol

The lock style implemented by the HP-UX network lock manager supports deadlock detection on a per-server basis only (see the *lockf(2)* and *fcntl(2)* in the *HP-UX Reference* for details).

Despite network lock manager's adherence to the *lockf()* / *fcntl()* semantics, a few subtle points about its behavior need to be mentioned. They are:

- When an NFS client goes down and comes back up, the lock managers on all servers are notified by their status monitors. The server lock managers release locks previously held at the request of the recovering client on the assumption that the client lock manager will request locks again when it wants them. When a server fails, however, the clients wait for the server to come back up. When it does, the server's lock manager gives the client lock managers a grace period to submit lock reclaim requests. During this period the server's lock manager accepts only reclaim requests from remote lock managers. The client status monitors notify their respective lock managers when the server recovers. The default grace period is 50 seconds.
- It is possible that, after a server failure, a client may not be able to recover a lock that it had on a file on that server. This can happen because another process may have accessed the lock before the recovering application process. In this case, the SIGLOST signal will be sent to the process (the default action for this signal is to kill the application).
- The local lock manager does not reply to the operating system's lock request until the server lock manager has acknowledged the local lock manager's request. Further, if the lock request is on a server new to the local lock manager, the lock manager registers its interest in that server with the local status monitor and waits for its reply. If either the status monitor or the server's lock manager are unavailable, the reply to a lock request for remote data is delayed until the server becomes available.
- Only advisory mode locking is supported. Enforcement mode is not supported for NFS files.

The Network Status Monitor

The Network Lock Manager relies heavily on the Network Status Monitor to maintain the inherently stateful locking service within the stateless NFS environment. However the status monitor can also be used to support other kinds of stateful network services and applications. Normally, failure recovery is one of the most difficult aspects of network application development, and requires a major design and installation effort. The status monitor simplifies this task.

The status monitor works by providing a general framework for collecting network status information. Implemented as a daemon that runs on all network computers, it uses a simple protocol that allows applications to monitor the status of other computers. Its use improves overall robustness, and avoids situations in which applications running on different computers (or even on the same computer) disagree about the status of a site - a potentially dangerous situation that can lead to inconsistencies in many applications.

Applications that use the status monitor do so by registering the computers they are interested in. The status monitor then tracks the status of those computers, and when one of them fails it notifies the interested applications of the failure, and the applications may take whatever actions are necessary to reestablish a consistent state.

A few advantages of this approach are:

- Only applications that use stateful services must pay the overhead - in time and in size - of dealing with the status monitor.
- The implementation of stateful network applications is simplified, since the status monitor shields application developers from the complexity of the network.

NIS Configuration and Maintenance

The Network Information Service (NIS) is an optional, distributed network lookup service that allows you to administer databases from one node on the network. (NIS was formerly known as Yellow Pages (YP), which is a registered trademark of British Telecommunications.) With NIS you can maintain a single set of user and group IDs for all nodes within a specified set (NIS domain).

The sections included in this chapter are:

- Key terms.
- NIS databases.
- Local and global maps.
- Escape sequences.
- Netgroups.
- Files related to NIS.
- NIS commands.
- NIS configuration.
- Verify NIS.
- NIS maintenance.

Refer to *ypfiles(4)* in the *HP-UX Reference* for a complete explanation of the NIS database and directory structure. If you do not have NIS administrative responsibilities, refer to the this manual for general NIS usage information.

Note If you configured the BIND name server, it will be used instead of NIS for host name and address resolution. However, NIS will still be used for all other information such as passwords. See *Configuring and Maintaining the BIND Name Server* chapter in the *Installing and Administering ARPA Services* manual.

For this chapter only, all references to servers and clients are NIS specific.

Key Terms

Key Terms	Definition
Bind	<ul style="list-style-type: none">- Process by which a client locates and directs all requests for data to a specific server.- Process of establishing the address of a socket that allows other sockets to connect to it or to send data to it. <p>Acronym for Berkeley Internet Name Domain. The BIND Name Server is a distributed network lookup service.</p>
Cluster	One or more workstations linked together with a local area network (LAN), but consisting of only one root directory. For more information on cluster concepts, see <i>Managing Clusters of HP9000 Computers: Sharing the HP-UX Filing System</i> .
Cluster Auxiliary Server	A cluster client with a disk drive that contains files shared by the other members of the cluster.
Cluster Client	A node in an HP-UX cluster that uses networking capabilities to share directories or files, but does not have a root directory directly attached. For HP-UX 8.0, cluster clients can have locally mounted disks for local data storage.
Cluster Root Server	The only node in an HP-UX cluster that has the root directory directly attached to it.
Escape Sequence (NIS)	Characters used within files to force inclusion and exclusion of data from NIS databases. The escape sequences are as follows: <ul style="list-style-type: none">* + (plus)* - (minus)* +@netgroup_name* -@netgroup_name
Export	To make a directory available to remote nodes via NFS.
File System	An entire unit (disk partition) that has a fixed size.
GID	A value that identifies a group in HP-UX.

Key Terms	Definition
Global (NIS)	A means of access in which the system always reads NIS maps rather than the local ASCII files.
Host	A node that has primary functions other than switching data for the network.
Internet Address	A four-byte quantity that is distinct from a link-level address and is the network address of a computer node. This address identifies both the specific network and the specific host on the network.
Key (NIS)	A string of characters (no imbedded blanks or tabs) that indexes the values within a map so the system can easily retrieve information. For example, in the <code>passwd.byname</code> map, the users' login names are the keys and the matching lines from <code>/etc/passwd</code> are the values.
Local (NIS)	A means of access in which the system first reads the local ASCII file. If it encounters an escape sequence, it then accesses the NIS databases.
Map (NIS)	<p>A file consisting of logical records; a search key and related value form each record. NIS clients can request the value associated with any key within a map.</p> <p>NIS map is synonymous with NIS database.</p>
Master Server (NIS)	The node on which one or more NIS maps are constructed from ASCII files. These maps are then copied to the NIS slave servers for the NIS clients to access.
Netgroup	A network-wide group of nodes and users defined in <code>/etc/netgroup</code> .
Network Information Service (NIS)	<p>An optional network service composed of databases (maps) and processes that provide NIS clients access to the maps. NIS enables you to administer these databases from one node.</p> <p>NIS may or may not be active; check with your system administrator.</p>
NIS Client	<ul style="list-style-type: none"> - A node that requests data or services from NIS servers. - An NIS process that requests other NIS processes to perform operations. <p><i>Note</i> An NIS client can also be configured as any combination of an NIS server, NFS client, or NFS server. An NIS server <i>must</i> also be configured as an NIS client.</p>

Key Terms	Definition
NIS Database	See "Map (NIS)."
NIS Domain	A logical grouping of NIS maps (databases) stored in one location. NIS domains are specific to NIS and are not associated with other network domains.
NIS Password	<p>The password for a user's login ID that exists in the NIS passwd map. The NIS password is the same one as the user password, but is administered through NIS.</p> <p>You do not have to have an NIS password to access the NIS databases.</p>
NIS Server	<ul style="list-style-type: none"> - A node that provides data (maps) or services to other nodes (NIS clients) on the network using NIS. - An NIS process that performs operations as requested by other NIS processes. <p><i>Note:</i> An NIS server <i>must</i> also be configured as an NIS client. It can also be configured as an NFS server, NFS client, or both.</p>
Propagate	To copy maps (data) from one NIS server to another.
Slave Server (NIS)	A node that copies NIS maps from the NIS master server and then provides NIS clients access to these maps.
UID	A value that identifies a user in HP-UX.
Value (NIS)	A unit of information stored in NIS maps; each value has a corresponding key (index) so the system can easily retrieve it. For example, in the passwd.byname map, the users' login names are the keys and the matching lines from /etc/passwd are the values.

NIS Databases

The ypmake script creates the standard NIS databases from the following ASCII files. You can also create additional NIS databases. (Refer to *ypfiles(4)* in the *HP-UX Reference*.)

/etc/group	/etc/passwd
/etc/hosts	/etc/protocols
/etc/netgroup	/etc/rpc
/etc/networks	/etc/services

Other maps may be present, like *ethers* and *mail.aliaes*, that may be used by other vendors or applications.

Note If the */usr/etc/yp* directory is part of a directory that supports only short file names (14 characters maximum), then any maps you create can have only 10 characters. This restriction exists because the *makedbm* command automatically adds the *.dir* and *.pag* suffixes to each map name. Refer to the *System Administration Tasks* manual for more information on short file name directories or files.

Local and Global Maps

Clients access the above ASCII files and their corresponding NIS maps in one of two ways, depending on whether the NIS maps are local or global.

- A map is *local* if the system first accesses the local ASCII file. If the file contains an escape sequence, the system then accesses the NIS database.
- A map is *global* if the system accesses only the NIS database (never accesses the local ASCII file).

If a node is not a client, the system accesses only the local ASCII files for information.

NIS Maps	Type	Access
/etc/group /etc/passwd	Local	<p>If a + (plus) entry exists at the beginning of a line, the system retrieves data from the corresponding NIS map; otherwise, the NIS maps are unused.</p> <p>Occurrences of +@netgroup_name and -@netgroup_name at the beginning of a line cause the system to reference NIS.</p> <p>(Refer to <i>group(4)</i> and <i>passwd(4)</i> in the <i>HP-UX Reference</i> for complete information regarding these escape sequences.)</p>
/etc/hosts /etc/netgroup /etc/networks /etc/protocols /etc/rpc /etc/services	Global	<p>The system consults only NIS for data. If NIS is not running, it looks at the local ASCII files.</p> <p>However, if the BIND Name Server is configured, the system will use it for host name and address resolution instead of NIS.</p>

Escape Sequences

Escape sequences are characters used within a file at the beginning of a line to force inclusion and exclusion of data from NIS databases. (Refer to *passwd(4)*, *hosts(4)*, *netgroup(4)*, *host.equiv(4)*, and *group(4)* in the *HP-UX Reference*.) The following table shows the escape sequences and their descriptions.

Escape Sequence	Description
+ (plus)	Use + (plus) in <i>/etc/passwd</i> and <i>/etc/group</i> to retrieve one or more entries from the NIS <i>passwd</i> and <i>group</i> maps, respectively. The plus designates specific entries to be retrieved from NIS.
- (minus)	Use - (minus) in <i>/etc/passwd</i> and <i>/etc/group</i> to ignore any subsequent entries with the same name. This process hides the matching names occurring in the NIS <i>passwd</i> and <i>group</i> maps, respectively. Therefore, it disallows access to particular entries.
+@netgroup_name	Use +@netgroup_name in <i>/etc/passwd</i> to insert the matching entries from the NIS <i>passwd</i> map for all members of a network group. <i>For ARPA Services:</i> Use +@netgroup_name in <i>/etc/hosts.equiv</i> and <i>\$HOME/.rhosts</i> to include a network group's entries from their lists of allowed users.
-@netgroup_name	Use -@netgroup_name in <i>/etc/passwd</i> to disallow the matching entries from the NIS <i>passwd</i> map for all members of a network group. <i>For ARPA Services:</i> Use -@netgroup_name in <i>/etc/hosts.equiv</i> and <i>\$HOME/.rhosts</i> to exclude a network group's entries from their lists of allowed users.

Netgroups

Netgroups are network-wide groups of nodes and users defined in `/etc/netgroup` on the master server. The primary purpose of `netgroup` is permission checking (allows/disallows) when executing remote mounts, logins(`rlogin`), and remote shell execution(`remsh`).

The following table shows the programs that consult the NIS netgroup maps and their descriptions. The parenthetical comments refer to sections in the *HP-UX Reference* where you can go for more information.

Program	Description
<i>login</i> (1)	Consults the maps to resolve netgroup names in <code>/etc/passwd</code> .
<i>mountd</i> (1M)	Consults the maps to resolve netgroup names in <code>/etc/exports</code> .
<i>rlogin</i> (1) <i>remsh</i> (1)	For <i>ARPA Services</i> : Consults the netgroup map if netgroup names are in <code>/etc/hosts.equiv</code> or <code>\$HOME/.rhosts</code> .

The controlling files for netgroup functionality are:

1. `/etc/netgroup`
2. `/etc/hosts.equiv`
3. `$HOME/.rhosts`

The master server uses `/etc/netgroup` to generate three NIS maps in the `/usr/etc/yp/domain_name` directory: `netgroup`, `netgroup.byuser`, and `netgroup.byhost`. The `netgroup` map contains basic information found in `/etc/netgroup`. The other two maps contain more specific information to accelerate the lookup of netgroups given the user or host.

To limit access to directories or files, edit `/etc/exports` to include the appropriate netgroup names. Then define the netgroup in `/etc/netgroup` using the following format. (Refer to *exports*(4) and *netgroup*(4) in the *HP-UX Reference*.)

Each line in the `netgroup` file define a group and the format:

```
groupname member1 member2 ...
```

MemberX is either another group name, or a triple. A triple being (hostname, username, domainname). If any of these fields are left empty, the empty field signifies a wild card which give access to hosts and users of that group. Field names that begin with something other than a letter, digit, or underscore do not match any value. For example:

```
alien3 (bishop,-,prison)
```

```
terminator (-,arnie,future)
```

The system `bishop` belongs to the group `alien3` in the domain `prison`, but no users belong to it. Similarly, the user `arnie` belongs to the group `terminator` in the domain `future`, but no systems belong to it.

Note The domain name filed must match your current domain name (as returned by the `domainname` command), or the entry will not be matched. Also, the user name filed is ignored for remote mounts. Only the hostname and domainname are used.

You can use the following conventions when editing the `/etc/exports` file:

- The memberX is equal to the triple (hostname, username, domainname).
- You can assign more than one triple to a netgroup by enclosing each separate set within parentheses (hostname, username, domainname).
- Leave any of these three fields empty to signify a wild card (i.e., blank fields match anything). For example, (`,,prison`) matches all hosts and users in the `prison` NIS domain.
- A minus (-) in any of these three fields means *match nothing*. For example, (`-,arnie,future`) does not match any hosts, but it does match the user `arnie` in the `future` NIS domain.
- Each hostname must have an entry in `/etc/hosts`. (See *hosts(4)* in the *HP-UX Reference*.)
- The *domainname* is the name of the NIS domain to which you currently belong. The commands using `/etc/netgroup` assume you are not looking for any NIS domain other than the one assigned on your node. (To list your current NIS domain name, execute the `domainname` command.)

EXAMPLE: The following example is a sample `/etc/netgroup` file. (Refer to `netgroup(4)` in the *HP-UX Reference* for a complete file format description and a definition of lines and fields.)

```
#
# sfmovies: Everyone, but bob, has a node.
# The node robocop does not have any users associated with it.
#
sfmovies alien3 recall
alien3 (woman,ripley,future) (android,bishop,future) (robocop,-,future)
recall (man,doug,future) (wife,laurie,future) (-,bob,future)
#
# terminator: Time-sharing on cyborg
#
terminator (cyborg,arnie,future) (cyborg,kyle,future) (cyborg,sarah,future)
#
# Others
#
allusers (-,future)
allhosts (-,future)
```

The NIS domain name for all the example netgroups is `future`. The following table shows how the users and hosts are classified into netgroups.

Netgroup	Users	Hosts
alien3	ripley, bishop	woman, android, robocop
recall	doug, laurie, bob	man, wife
sfmovies	ripley, bishop, doug, laurie, bob	woman, android, man, wife, robocop
terminator	arnie, kyle, sarah	cyborg
allusers	every user in the NIS passwd map	no hosts
allhosts	no users	all hosts in the NIS hosts map

Files Related to NIS

For ARPA Services: The files `/etc/hosts.equiv` and `$HOME/.rhosts` are not in the NIS system; however, they are related to NIS. If these files contain a plus (+) or minus (-) entry with the argument `@netgroup`, the system consults the NIS `netgroup` map for data. (Refer to `netgroup(4)` and `hosts.equiv(4)` in the *HP-UX Reference*.) For example, in `/etc/hosts.equiv` a line consisting of:

```
+@sfmovies
```

will include all members (subgroups and their triples) of `sfmovies` netgroup defined in the local file `/etc/netgroup` or in the NIS database. A line consisting only of a plus (+) allows access to all hosts.

Note Before implementing this line, you must define the netgroups then define the subgroups which contain the triples.

An inappropriate entry in the `hosts.equiv` file would be:

```
+@alien3 +@robocop
```

Both are subgroups, not the defined netgroup. A correct entry would be:

```
@sfmovies +@rekall
```

`sfmovies` has been defined as the netgroup and `rekall` was defined as the subgroup.

The same holds true for `$HOME/.rhosts`. In `$HOME/.rhosts`, a host name followed by a plus (+) means any user coming from that host name will be allowed to access this account through `rlogin` or `remsh`. (See `hosts.equiv(4)` in the *HP-UX Reference*.) But there is one major difference, if any user has a login on a system they can get trusted access by making an entry in their `$HOME/.rhost` file on the remote system. This happens even if they are a disallowed user in any of the following scenarios:

1. They are not in any allowed netgroup.
2. They are in a disallowed netgroup. (i.e., `~@netgroup ~@subgroup`).

The system calls `gethostent`, `gethostbyname`, and `gethostbyaddr`. All these intrinsics go to the `.rhosts` by default even if you are running NIS or any named systems. Any user can access a trusted system in spite of the netgroups maps. `.rhostsfiles` can override netgroup functionality.

To avoid this override, be certain:

1. There are no hidden .rhost files within the netgroups you define. The remote commands rcp, remsh, rlogin, rlogind, and remshd use this file as the default. ruserok(3M) first checks the /etc/hosts.equiv then checks the \$HOME/.rhosts by default when attempting to validate a user.
2. Do not use netgroup entries in the .rhost file. Matching that takes place in this file is done literally.
3. If a user is a superuser, the .rhost file is the only file consulted. The hosts.equiv is bypassed completely.

NIS Commands

Refer to the following table for a brief description of all NIS commands. Refer to the Common Commands chapter in this manual for a more detailed description of the NIS commands you might want to use on a daily basis (i.e., those NIS commands that do not require superuser access). The parenthetical comments refer to sections in the *HP-UX Reference* where you can go for more information.

NIS Commands	Description
<i>domainname</i> (1)	Use <i>domainname</i> to determine or change an NIS domain name.
<i>makedbm</i> (1M)	<i>Note:</i> Use this version of <i>makedbm</i> only with NIS. A tool for building NIS maps. Use <i>makedbm</i> to build or rebuild databases not built by <i>/usr/etc/yp/ypmake</i> . Use <i>makedbm</i> to disassemble a map so that you can see the key-value pairs comprising it.
<i>ypbind</i> (1M)	Used by each client to determine to which server it should bind. It does have options which allow you to enable/disable remote <i>ypset</i> commands. <i>Note:</i> This entry exists in the <i>HP-UX Reference</i> as <i>ypserv</i> (1M); it exists online as <i>ypbind</i> (1M).
<i>ypcat</i> (1)	Lists the contents of an NIS map.
<i>ypinit</i> (1M)	On NIS master servers, <i>ypinit</i> constructs maps from <i>/etc</i> files. On NIS slave servers, <i>ypinit</i> copies the initial map versions from the master server.
<i>ypmake</i> (1M)	A script, initially called by <i>ypinit</i> , that builds standard NIS maps from ASCII files. These files are usually in <i>/etc</i> : <i>passwd</i> , <i>hosts</i> , <i>group</i> , <i>netgroup</i> , <i>networks</i> , <i>protocols</i> , <i>rpc</i> , and <i>services</i> .
<i>ypmatch</i> (1)	Prints the value for one or more specified keys in an NIS map.

NIS Commands	Description
<i>yppasswd</i> (1)	Changes the password for your current login ID in the NIS passwd map. (You do not have to have an NIS password to access the NIS databases.)
<i>yppasswd</i> (1M)	A server, running only on the master server, that permits users to change their password in the NIS password map.
<i>yppoll</i> (1M)	Asks any ypserv for the information it holds about a single map.
<i>yppush</i> (1M)	Used by the master server to administer a running NIS system. The yppush command causes an NIS map to be copied (using ypxfr) from the maps' master server to each slave server in the NIS domain.
<i>ypserv</i> (1M)	Provides access to data stored in NIS maps on servers. If operating in an HP-UX cluster environment, ypserv should be running on the root server.
<i>ypset</i> (1M)	Tells the ypbind process to obtain NIS for an NIS domain from a specific server. When used with the -h option (Remote Nodes), ypbind, on that node must be running with options that allow it to accept requests from ypset.
<i>ypwhich</i> (1)	Tells you which server a node is currently using or which server is master of a specified map.
<i>ypxfr</i> (1M)	Transfers an NIS map from one slave server to another. Run ypxfr <i>one</i> of three ways: - yppush periodically. - ypxfr interactively. - via cron periodically.

NIS Configuration

The NIS configuration covered in this section include:

1. Compare `/etc/newconfig` files to existing files.
2. Create an NIS master server.
3. Create an NIS client.
4. Create an NIS slave server.
5. Propagate the NIS maps.
6. Verify NIS.

1. Compare `/etc/newconfig` Files to Existing Files

When you installed the NFS services software, several new files were copied into the `/etc/newconfig` directory. Perform the following steps to prepare to configure NIS.

1. Compare each `/etc/newconfig` file listed below with its counterpart shown in the following table.

File in <code>/etc/newconfig</code> directory	Counterpart in <code>/usr/etc/yp</code> directory
<code>ypinit</code>	<code>ypinit</code>
<code>yp_Makefile</code>	<code>Makefile</code>
<code>ypmake</code>	<code>ypmake</code>
<code>ypxfr_1perday</code>	<code>ypxfr_1perday</code>
<code>ypxfr_1perhour</code>	<code>ypxfr_1perhour</code>
<code>ypxfr_2perday</code>	<code>ypxfr_2perday</code>

2. If the files are the same, skip to the next section, 2. Create an NIS Master Server.

3. If you have previously customized the files that exist in the `/usr/etc/yp` directory, or if the files are from an older release of the software, they will differ from files in `/etc/newconfig`. If there are differences, copy the current files in `/usr/etc/yp` to a safe location and do *one* of the following:
 - Change the versions in `/usr/etc/yp` to reflect the differences in the files in `/etc/newconfig`.
 - Copy the files in `/etc/newconfig` to `/usr/etc/yp`. Then re-customize the files in `/usr/etc/yp` if necessary.

2. Create an NIS Master Server

You must be superuser to create an NIS master server (i.e., to build the NIS master databases). You should also be in a single user state of operation.

An NIS server *must* also be configured as an NIS client. It can also be configured as an NFS server, NFS client, or both.

Preparations for Creating an NIS Master Server

Perform the following steps before creating your master server:

1. Ensure `/etc` files are complete and current: `passwd`, `hosts`, `group`, `networks`, `protocols`, `rpc`, and `services`.
2. If you know the correct configuration, create the `/etc/netgroup` file. (See `netgroup(4)` in the *HP-UX Reference*.)

Note The NIS maps store only the first occurrence if:

- A duplicate user name or duplicate user ID exists in `/etc/passwd`.
 - A duplicate internet address or duplicate host name exists in `/etc/hosts`.
-

Restricting Access to the Master Server

If you want to restrict access to the master server to a smaller set of users than defined by the complete `/etc/passwd` file, perform the following steps:

1. Copy the entire `/etc/passwd` file to a different file (e.g., `/etc/passwd.nis`).
2. Delete undesired users from the original `/etc/passwd` file. To prevent all entries in the NIS `passwd` map from being able to log in, this smaller file *should not* include the following line:

```
+ ::0:0:::
```

3. Edit `/usr/etc/yp/ypinit` as follows:

```
CHANGE: PWFIL = /etc/passwd
```

```
TO: PWFIL = /etc/passwd.nis
```

4. Edit `/etc/netnfsrc` as follows:

```
CHANGE: /usr/etc/rpc.yppasswdd /etc/passwd -m passwd PWFIL = /etc/passwd
```

```
TO: /usr/etc/rpc.yppasswdd /etc/passwd.nis -m passwd PWFIL = /etc/passwd.nis
```

5. If you have `rpc.yppasswdd` running, kill and restart it.

```
/usr/etc/rpc.yppasswdd /etc/passwd.nis -m passwd PWFIL = /etc/passwd.nis
```

If in the future you need to run `ypmake` and you have restricted access to the master server as just described, enter the following line:

```
/usr/etc/yp/ypmake passwd PWFIL = /etc/passwd.nis
```

Note For information on C2 Security, refer to the *HP-UX System Security Manual, A Beginner's Guide to Using Shells*, and the *HP-UX Beginner's Guide*.

Creating an NIS Master Server

Perform the following steps to create your master server:

1. Set the NIS domain name using the `domainname` command. This NIS domain name must be the same one used for all clients and servers within this NIS domain as shown in the example:

```
domainname nis_domain_name
```

2. Execute `ypinit` with the `-m` parameter in one of two ways:

- If you want to make this node a master server of the domain name that you set in Step 1, enter:

```
/usr/etc/yp/ypinit -m
```

- If you want to make this node a master server of a different domain name than the one you set in Step 1, enter:

```
/usr/etc/yp/ypinit -m [ DOM = XXX ]
```

XXX represents the domain name for which you are setting this node to be a master server.

3. The system asks whether you want the procedure to quit at the first non-fatal error. Do *one* of the following:
 - Respond `no` or `n` for `ypinit` to continue regardless of the errors. After the procedure finishes, correct all errors that occurred.
 - Respond `yes` or `y` for `ypinit` to quit at the first error. Correct each error as it occurs. This procedure takes longer since you have to correct the errors one by one and run `ypinit` until no more errors occur.
4. The `ypinit` script prompts you for a list of hosts that will become servers.

Starting the NIS Master Server

You should edit `/etc/netnfsrc` to automatically start the master server at boot time. You can also manually start it now.

Manually Starting NIS Master Server	Automatically Starting NIS Master Server (at Boot Time)
<p>1. If you have not already done so, set the NIS domain name using the <code>domainname</code> command. This NIS domain name must be the same one used for all clients and servers within this NIS domain.</p> <pre>domainname nis_domain_name</pre> <p>2. Execute <code>ypserv</code>:</p> <pre>/usr/etc/ypserv</pre> <p><i>Note:</i> If operating in an HP-UX cluster environment, start <code>ypserv</code> only on the node that you wish to make the master server, and start <code>ypbind</code> on every other node.</p> <p>3. Execute <code>ypbind</code>:</p> <pre>/etc/ypbind</pre>	<p>1. Go into <code>/etc/netnfsrc</code>.</p> <p><i>Note:</i> A zero in the <code>NIS_CLIENT</code>, <code>NIS_MASTER_SERVER</code>, or <code>NIS_SLAVE_SERVER</code> field disables the node from working as a client, master server, or slave server respectively.</p> <p>2. Set <code>NISDOMAIN</code> to the NIS domain name.</p> <pre>NISDOMAIN = nis_domain_name</pre> <p>You will need to use this same NIS domain name for all clients and servers within this NIS domain.</p> <p>3. Set <code>NIS_MASTER_SERVER</code> to a value other than zero. Changing this variable permits users to change their NIS password.</p> <pre>NIS_MASTER_SERVER = 1</pre> <p>4. Set the <code>NIS_SLAVE_SERVER</code> to zero to disable the node as a slave server.</p> <pre>NIS_SLAVE_SERVER = 0</pre> <p>5. Set <code>NIS_CLIENT</code> to a value other than zero.</p> <pre>NIS_CLIENT = 1</pre>

3. Create an NIS Client

You must be superuser to create an NIS client.

An NIS client can also be configured as an NFS client, NFS server or both. All NIS servers *must* also be configured as NIS clients. Before creating an NIS client you must:

1. Determine an NIS domain on your network for the client you intend to create.
2. Ensure that a server is available in the NIS domain in which the client will exist (i.e., NIS databases exist and ypserv is running). (Refer to the section 2. Create an NIS Master Server.) If a server is not available in the same NIS domain as the client, users will be unable to log into the client.

Creating an NIS Client

Customize the following files that traditionally store the information. (For suggested modifications, refer to the following section Altering a Client's Files.)

Note *Do not* abbreviate or eliminate these files if the client is also the master server.

/etc/group	/etc/passwd
/etc/hosts	/etc/protocols
/etc/netgroup	/etc/rpc
/etc/networks	/etc/services

Altering a Client's Files

The following table provides suggestions for altering the client files.

Client File	Suggested Modification
/etc/group	<p>You may want to reduce /etc/group to a single line containing a plus (+) followed by a colon (:) or simply place the line with + as the first line of this file. This line forces all translations of group names and group IDs to occur via NIS since group is a local map.</p> <p>+:</p>
/etc/hosts	<p>Ensure /etc/hosts contains an entry for the local host name. The system accesses these entries when NIS is not yet available. After the ypbind process is running, the system never accesses /etc/hosts.</p> <p>EXAMPLE: Sample NIS client's /etc/hosts entry</p> <pre>192.9.1.87 local_host # Byron W. Donnell</pre> <p><i>Note:</i> If you configured the BIND name server, it will be used instead of NIS for host name and address resolution. However, NIS will still be used for all other information such as passwords. See <i>Configuring and Maintaining the BIND Name Server</i> chapter in the <i>Installing and Administering ARPA Services</i> manual.</p>

Client File	Suggested Modification
<p><code>/etc/hosts.equiv</code> (For <i>ARPA Services</i>)</p>	<p>The system first accesses <code>/etc/hosts.equiv</code> directly. If a <code>+@netgroup</code> or <code>-@netgroup</code> entry exists, the system accesses the NIS netgroup map.</p> <p><i>Note:</i> Using netgroup reduces <code>rlogin</code> and <code>remsh</code> problems that occur because different <code>/etc/hosts.equiv</code> files exist on different nodes.</p> <p>For more control over logins, edit <code>/etc/hosts.equiv</code> as follows:</p> <ol style="list-style-type: none"> 1. Enter either a plus (+) or (-) to enable (trusted) or disable (distrusted) login, respectively. 2. Enter the at (@) character. 3. Enter the name of the netgroup as defined in the global netgroup database. <p>EXAMPLE:</p> <pre>+@netgroup1 (trusted) -@netgroup2 (distrusted)</pre>
<p><code>\$HOME/.rhosts</code> (For <i>ARPA Services</i>)</p>	<p>The system first accesses <code>\$HOME/.rhosts</code> directly. If a <code>+@netgroup</code> or <code>-@netgroup</code> entry exists, and has no hidden <code>.rhost</code> files the system accesses the NIS netgroup map. (Refer to the previous section, Netgroups, in this chapter.)</p> <p>Since the superuser's <code>\$HOME/.rhosts</code> controls remote superuser access to the local node, HP recommends restricted access. To restrict access, either make the list of trusted hosts explicit or use netgroup names.</p>

Client File	Suggested Modification
/etc/passwd	<p>Ensure /etc/passwd contains:</p> <ul style="list-style-type: none"> - Entries for the root user. - Entries for the primary users. - An escape entry to use NIS. <p>Entries in the local /etc/passwd file mask identical name entries in the NIS passwd maps. Delete all other names and enter +::0:0:: as the last line. This line causes library routines looking for a particular entry to search the NIS database:</p> <p style="padding-left: 40px;">+::0:0::</p> <p>EXAMPLES: Sample entries in /etc/passwd</p> <p style="padding-left: 40px;">+ap:::Dave Hamil:/usr2/ap:/bin/csh</p> <p>The system pulls an entry for ap from the NIS passwd5D map because of the + (plus) escape character.</p> <p>It obtains the UID, GID, and password from NIS and obtains the comment field, home directory, and default shell from the /etc/passwd entry.</p> <p>If no entry for ap exists in NIS, the system reacts as though no entry for ap exists anywhere:</p> <p style="padding-left: 40px;">ap::140:100:Mike Donn:/usr2/ap:/bin/csh</p> <p>Since the plus (+) escape character is not present, the system does not access NIS. User ap has no password.</p>

Client File	Suggested Modification
<p><code>/etc/passwd</code> (continued)</p>	<p>EXAMPLES: Sample entries in <code>/etc/passwd</code></p> <p>+ ap:</p> <p>The system obtains all information from the NIS <code>passwd</code> map for user <code>ap</code>.</p> <p>+ ::0:0::</p> <p>The system obtains all information from the NIS <code>passwd</code> map for all users not already encountered.</p> <p>If the entry is not found in the NIS database, the search continues in the <code>/etc/passwd</code> file.</p>

Starting the NIS Client

You should edit `/etc/netnfsrc` to automatically start the client at boot time. You can also manually start it now.

Manually Starting NIS Client	Automatically Starting NIS Client (at Boot Time)
<p>1. If you have not already done so, set the NIS domain name using the <code>domainname</code> command. This NIS domain name must be the same one used for all clients and servers within this NIS domain.</p> <p><code>domainname nis_domain_name</code></p> <p>2. Execute <code>ypbind</code>.</p> <p><code>/etc/ypbind -ypsetme</code></p>	<p>1. Go into <code>/etc/netnfsrc</code>.</p> <p>2. Set <code>NISDOMAIN</code> to the same NIS domain name used on all clients and servers within this NIS domain.</p> <p><code>NISDOMAIN = nis_domain_name</code></p> <p>3. Set <code>NIS_CLIENT</code> to a value other than zero.</p> <p><code>NIS_CLIENT = 1</code></p> <p><i>Note:</i> A zero in the <code>NIS_CLIENT</code> field disables the node from working as an NIS client.</p>

Note If you want the node to be a server also, refer to either the section 2. Create an NIS Master Server or 4. Create an NIS Slave Server for complete instructions.

4. Create an NIS Slave Server

You must be superuser to create an NIS slave server.

You may want to create slave servers to improve the reliability of your system. An NIS server *must* be configured as an NIS client. It can also be configured as an NFS server, NFS client, or both.

Preparations for Creating an NIS Slave Server

Before creating a slave server, ensure the following:

- The master server exists (see 2. Create an NIS Master Server section).
- The ypserv daemon is running on the master server.

Note If you are operating in an HP-UX environment, HP recommends that only one cnode per cluster should be an NIS server.

Creating an NIS Slave Server

Refer to the following steps to create a slave server:

1. Set the NIS domain name using the `domainname` command. This NIS domain name must be the same one used for all clients and servers within this NIS domain:

```
domainname nis_domain_name
```

2. Execute `ypinit` with the `-s` parameter in one of two ways:

- If you want to make this node a slave server of the domain name that you set in Step 1, enter:

```
/usr/etc/yp/ypinit -s master_server_name
```

- If you want to make this node a slave server of a different domain name than the one you set in Step 1, enter:

```
/usr/etc/yp/ypinit -s master_server_name [ DOM = XXX ]
```

XXX represents the domain name for which you are setting this node to be a slave server.

3. The system asks whether you want the procedure to quit at the first non-fatal error. Do *one* of the following:
 - Respond **no** or **n** for **ypinit** to continue regardless of the errors. After the procedure finishes, correct all errors that occurred.
 - Respond **yes** or **y** for **ypinit** to quit at the first error. Correct each error as it occurs. This procedure takes longer since you have to correct the errors one by one and run **ypinit** until no more errors occur.
4. Since the slave server is also a client, customize the files which traditionally implement the database. Refer to the previous table *Altering an NIS Client's Files* in the section 3. Create an NIS Client.

Starting the NIS Slave Server

You should edit `/etc/netnfsrc` to automatically start the slave server at boot time. You can also manually start it now.

Manually Starting NIS Slave Server	Automatically Starting NIS Slave Server(at Boot Time)
<p>1. If you have not already done so, set the NIS domain name using the <code>domainname</code> command. This NIS domain name must be the same one used for all clients and servers within this NIS domain.</p> <pre>domainname nis_domain_name</pre> <p>2. Execute <code>ypserv</code>.</p> <pre>/usr/etc/ypserv</pre> <p><i>Note:</i> If operating in an HP-UX cluster environment, start <code>ypserv</code> only on a single node, and start <code>ypbind</code> on every other cnode.</p> <p>3. Execute <code>ypbind</code>.</p> <pre>/etc/ypbind -ypsetme</pre>	<p>1. Go into <code>/etc/netnfsrc</code>.</p> <p><i>Note:</i> A zero in the <code>NIS_CLIENT</code>, <code>NIS_MASTER_SERVER</code>, or <code>NIS_SLAVE_SERVER</code> field disables the node from working as a client, master server, or slave server respectively.</p> <p>2. Set <code>NISDOMAIN</code> to the same NIS domain name used on all clients and servers within this NIS domain.</p> <pre>NISDOMAIN = nis_domain_name</pre> <p>3. Set the <code>NIS_MASTER_SERVER</code> to zero to disable the node as a master server.</p> <pre>NIS_MASTER_SERVER = 0</pre> <p>4. Set <code>NIS_SLAVE_SERVER</code> to a value other than zero.</p> <pre>NIS_SLAVE_SERVER = 1</pre> <p>5. Set <code>NIS_CLIENT</code> to a value other than zero.</p> <pre>NIS_CLIENT = 1</pre>

5. Propagate NIS Maps

"Propagate NIS maps" means to copy a map from one server to another. Initially, `ypinit` copies the maps when you create slave servers. After the slave servers are initialized, you will use `ypxfr` to transfer updated maps from the master server to the slaves. You can run `ypxfr` three ways:

- Periodically from `cron` on each slave server.
- Periodically by executing `yppush` on the master server.
- Interactively executing `ypxfr` on a slave server.

`crontab`

Maps have different change rates. For example, `protocols.byname` may not change for months, but `passwd.byname` may change several times a day.

Create `crontab` entries to periodically run `ypxfr` at a rate appropriate for each map in the NIS database. The `ypxfr` command will contact the master server and transfer the map only if the master's copy is more recent than the local copy.

To avoid a `crontab` entry for each map, group the maps with approximately the same change characteristics. Place these maps in a shell script you can run via `cron`. Suggested groupings, mnemonically named, are in `/usr/etc/yp`: `ypxfr_1perhour`, `ypxfr_1perday`, and `ypxfr_2perday`. If the rates of change are inappropriate for your needs, either modify or replace these shell scripts.

Execute these shell scripts on each slave server in the NIS domain. Alter the exact time of execution from one server to another to prevent this process from slowing down the master.

EXAMPLE: crontab entries for using these scripts

```
# At 9:00 PM daily, transfer the group, networks, protocols,  
# rpc, services, and ypservers maps.
```

```
0 21 * * * /usr/etc/yp/ypxfr_1perday
```

```
# At 45 minutes past the hour, transfer the passwd maps.
```

```
45 * * * * /usr/etc/yp/ypxfr_1perhour
```

```
# At 11:30 AM and 11:30 PM daily, transfer the ethers,  
# hosts,mail.aliases and netgroup maps.
```

```
30 11,23 * * * /usr/etc/yp/ypxfr_2perday
```

You can check and transfer maps with unique change characteristics by explicitly invoking `ypxfr` from within your crontab file.

EXAMPLE: `25,55 * * * * /usr/etc/yp/ypxfr passwd.byname`

yppush

Execute `yppush` only on the master server to copy a map to each server in the NIS domain (retrieved from the `ypservers` map).

1. The `yppush` command sends a “transfer map” request to each of the servers.
2. In turn, `ypserv` on each server executes `ypxfr -C`.
3. The `ypserv` daemon then passes `ypxfr` the information needed to identify and transfer the map.

EXAMPLE: `/usr/etc/yp/yppush passwd.byname`

If you wish to run multiple `ypxfrs` at a time and control the timeout value of these transfers, use the `-m` and `-t` options. For information about these options, see `yppush(1M)` in the *HP-UX Reference*.

ypxfr

Execute **ypxfr** interactively only on the slave servers and only in exceptional situations. For example, execute it when creating a temporary server to make a test environment, or when trying to quickly propagate maps to make a slave server consistent with the other slave servers.

EXAMPLE: `/usr/etc/yp/ypxfr map_name`

If you want the map transferred from a server other than the master server, specify it using the `-h` option with **ypxfr**.

EXAMPLE: `/usr/etc/yp/ypxfr -h server_name passwd.byname`

6. Verify NIS

To verify a client is bound to a server, log into that client and execute `ypwhich`. *One* of the following will occur:

- If the client is bound, the response will be the host name of that server.
- If the client is not bound, you will receive the following message.

NIS domain *domain_name* not bound.

If you try `ypwhich` several times and continue to receive the not bound response, the node is unable to locate a server for that NIS domain on the network. Review your NIS configuration process. If you did not make errors, refer to the Troubleshooting chapter.

To verify that NIS is being accessed, log into a client node as a user whose password entry must be served by NIS. If the login does not work, review your NIS configuration process. If you did not make errors, refer to the Troubleshooting chapter.

You have now completed configuring NIS. Do *one* of the following:

- If you are configuring NIS for the first time (with NFS Services), and you plan to use the Virtual Home Environment (VHE), you can now skip to the VHE Configuration and Maintenance chapter.
- If you do not plan to use VHE, execute `/etc/netnfsrc` to complete the configuration procedure.

NIS Maintenance

To keep NIS running correctly and efficiently, ensure it stays configured to meet your changing needs. Refer to the following sections to help you meet these needs:

- Disable NIS.
- Modify NIS maps.
- Add new NIS servers.
- Add new users to a node.
- Make a different node the NIS master.
- Change NIS password.
- Log files.
- Create non-standard NIS maps.

Disable NIS

You must be super user to disable NIS. If you choose to disable NIS, do the following:

1. Set the NIS domain name to null (no spaces within double quotes).

```
domainname ""
```
2. If NIS is currently running, kill the `ybind` and `ypserv` processes.
3. Edit `/etc/netnfsrc` to change the NIS values:
 - a. Change the `NIS_MASTER_SERVER`, `NIS_SLAVE_SERVER`, and `NIS_CLIENT` values to zero:

```
NIS_MASTER_SERVER=0  
NIS_SLAVE_SERVER=0  
NIS_CLIENT=0
```
 - b. Remove the `NISDOMAIN` variable if one exists:

```
NISDOMAIN =
```
4. If the above NIS domain is specified in `/etc/netgroup`, remove the NIS domain name throughout `/etc/netgroup`.

5. Restore any files that you altered for NIS use. For example, you may need to add users back to the `/etc/passwd` file.
6. Reboot the system.

Modify NIS Maps

You must be superuser to modify NIS maps.

Caution Modify maps only on the master server; otherwise, the changes will not be propagated correctly to the slave servers.

You may change most of the standard NIS maps, like `/etc/hosts`, by first editing the ASCII file and then running `ypmake`. Refer to the following Manual Modifications to NIS Maps section if you are:

- Adding non-standard maps.
- Editing maps for which no ASCII file exists.
- Changing the set of servers after the system is running.

Whether using `ypmake` in `/usr/etc/yp` or one of the following manual procedures, the goal is the same; a new, well-formed database must reside in the NIS domain directory on the master server. (Refer also to *makedbm(1M)* in the *HP-UX Reference*).

Caution Never modify a map directly; always use `makedbm` to create the map.

Manual Modifications to NIS Maps

You may want to change the following maps manually:

- Non-standard maps (i.e., those that are specific to the applications of a particular vendor or site, but are not part of HP's release).
- Maps that rarely change
- Maps for which no ASCII file exists (e.g., ypservers map).

To make a change, do the following:

1. Change to the directory of the maps you want to modify:

```
cd /user/etc/yp/nis_domain_name
```

2. Execute `makedbm -u` to disassemble the map into a form which is modifiable using HP-UX tools:

- a. Redirect the `makedbm -u` output to a temporary file and modify it. Execute `makedbm` using the temporary file as input to create the new versions.

EXAMPLE:

```
../makedbm -u mapname > tmpfile  
vi tmpfile # (make the required changes)  
../makedbm tmpfile mapname  
rm tmpfile
```

- b. Use a pipe to modify the `makedbm` output which you can then direct as input to `makedbm`. *Note:* You can use this method only if the disassembled map is updated via `awk`, `sed`, or a `cat` append.

EXAMPLE: Add a new key-value pair to the `map_name` map

```
(../makedbm -u map_name; echo newkey newvalue) | ../makedbm - map_name
```

Examples for Creating Non-Standard NIS Maps

Suppose you want to create a non-standard NIS map. You want it to consist of key-value pairs in which the keys are strings like al, bl, cl, and dl, and the values are ar, br, cr, and dr. After creating the map, you notice it is missing dl and dr.

You could use *one* of two procedures to create the new map: one using an existing ASCII file, the other using standard input.

Example for Existing ASCII File

Assume the following situation:

- An ASCII file exists named `/usr/etc/yp/john_map.asc`.
- The file was created with an editor or shell script on the master server.
- `john_map` is the name of the map you want to recreate.
- `graphs_domain` is the NIS domain subdirectory where the map is located.
- The NIS map was created from this file by entering:

```
cd /usr/etc/yp
./makedbm john_map.asc graphs_domain/john_map
```

Now you notice the map is missing dl and dr. To correct the error, modify the map by first modifying the ASCII file as follows:

```
cd /usr/etc/yp
< make editorial change to john_map.asc to add the dl and dr line >
./makedbm john_map.asc graphs_domain/john_map
```

To verify the new map has the changes you made, enter the following command:

```
./makedbm -u graphs_domain/john_map | more
```

Example: Using Standard Input

Assume the following situation:

- `wes_map` is the name of the map you want to create (no ASCII file exists from which the map was built).
- `reports_domain` is the NIS domain subdirectory in which you will create the map.

First, create the NIS map from the keyboard by entering input on the master server as follows:

```
cd /usr/etc/yp
./makedbm - reports_domain/wes_map
al ar
bl br
cl cr
[CTRL]-[D]
```

To modify the map, use `makedbm` to create a temporary ASCII intermediate file that can be edited:

```
cd /usr/etc/yp
./makedbm -u reports_domain/wes_map > wes_map.temp
```

Now edit `wes_map.temp` to add the `dl` and `dr` line. Create a new version of the database with the following commands:

```
./makedbm wes_map.temp reports_domain/wes_map
rm wes_map.temp
```

Add or Delete a NIS Server

You must be superuser to add new NIS servers.

If a new slave server is not in the original set, recreate the `ypservers` map on the master server. If needed, rebuild the `hosts` map (refer to `ypmake(1M)` in the *HP-UX Reference*):

1. If the server's address is not in `/etc/hosts`, edit `/etc/hosts` to include the new server's address and then execute `ypmake`:

```
< Edit /etc/hosts >
/usr/etc/yp/ypmake hosts
```

2. Add or delete the host's name to or from the ypservers map in the NIS domain as shown in the following example. Do not delete the master server from the list.

```
cd /usr/etc/yp  
./makedbm -u nis_domain_name/ypservers >/tmp/nis_server_list
```

<Edit /tmp/nis_server_list. Add or delete any slave server >

```
./makedbm /tmp/nis_server_list nis_domain_name/ypservers  
./yppush ypservers  
/bin/rm /tmp/nis_server_list
```

3. If you added a slave server, complete the steps in the section 3. Create a NIS Slave Server.

Add New Users to a Node

You must be superuser to add new users to a node.

Refer to the *System Administration Tasks* manual to add new users to a node. The procedure consists of:

1. Editing the master server's /etc/passwd and /etc/group files.
2. Making a home directory.
3. Defining the new user's environment.

Remember to update the NIS passwd and group databases by running /usr/etc/yp/ypmake. If you are using an alternate file to build the NIS password databases, use its full path name instead of /etc/passwd:

```
/usr/etc/yp/ypmake group passwd PWFIL = alternate_passwd_file
```

Make a Different Node the NIS Master

You must be superuser to change the NIS master server to a different node.

1. Copy the following files from your current master server to the node that will be the new master server:

- /etc/hosts
- /etc/netgroup
- /etc/networks
- /etc/protocols
- /etc/rpc
- /etc/services

2. Kill the `rpc.yppasswdd` process on the current master server.
3. Merge `/etc/group` and `/etc/passwd` on the current master server with those on the node that will be the new master server. (If using an alternate password file, you need only copy it.) This merging creates files suitable for building maps for all clients.

Merging ensures machine-specific password and group entries are kept intact. Either save or delete entries taken from the old master server files. For example, in `/etc/passwd` save user entries and remove the other node's root entry.

4. If `/usr/etc/yp/ypmake`, `/usr/etc/yp/ypinit`, or `/usr/etc/yp/Makefile` was modified on the old master server to build non-standard maps, copy them and other files from which the non-standard maps are built.
5. On the new master server, complete all steps in the 1. Create a NIS Master Server section.
6. To prevent starting `yppasswdd` on the old master server, edit its `/etc/netnfsrc` file to change the `NIS_MASTER_SERVER` value to zero:

```
NIS_MASTER_SERVER=0
```

7. If the old master server is to be a slave server, complete the steps in the 3. Create a NIS Slave Server section and the steps in the 2. Create a NIS Client section.
8. Reboot the new master server.
9. Reboot the old master server.
10. To ensure maps are consistent on all servers, execute `ypinit` on each slave server using the new master server's host name:

```
ypinit -s new_master_hostname
```

Create or Change NIS Password

The NIS password is the password for a user's login ID that exists in the NIS `passwd` map. The NIS password is used as the user password, but is administered through NIS. Note, you do not have to have a NIS password to access the NIS database.

If you change your password with the `passwd` command, you will change only the entry in your local `/etc/passwd` file if the entry exists. If your password is not in the file, the following error message occurs.

Permission denied.

If this error occurs, or if you would like to change your password while NIS is in use, execute `yppasswd`.

NIS Password Installation Guidelines

There are several items you should be aware of to be able to do a correct installation of `yppasswd`.

- You may have slave servers which only get updated at regular intervals and which are used by clients in the same domain (i.e., once/hour, once/day, etc.). Until a slave server receives a new `passwd` map from the master server a new password will not be activated. The user will be prompted for a new change until the new map is copied to the slave server. It is recommended that the `yppasswd` maps be copied to the slave servers at rates which minimize login delays to an acceptable level.
- There is a small delay between when the `passwd` daemon updates the `/etc/passwd` file on the master server and when `ypmake` completes. During this delay, the user cannot login when the client is using NIS. This delay is unavoidable because it takes a short time to run `ypmake` to update the `passwd` map.

The delay is dependent on the master server's load. Keeping this in mind, it is probably not a good idea to change everyone's `passwd` aging flag at the same time on the same day. This will cause system overload and excessive delays in productivity for the system users who are waiting to log on to the system after being forced to change their passwords due to the `passwd` aging flag expiring their old ones. This is especially the case for those users who rely on slave servers for updated maps in order to log in.

To avoid system overloads and delays in productivity, you as the system administrator, can install a `crontab` file on the slave server. The `crontab` file will execute `ypxfer` at frequent intervals, which update only the `passwd` maps and if the slave server already has a current map, no transfer is made. Rather than using the master server to update the maps, the slave servers

determine if they need to be updated, decreasing system overloads. The rate of these intervals can be set for once a minute or more depending on system demands.

There is a sample script for password maps that may be used. They are available in the `/usr/etc/yp` file. The script is called `ypxfr_1perhour`. You can use this script as a template to create a `crontab` file for the time values you think are appropriate for your system.

NIS Password Guidelines

The following list provides the requirements for creating and changing NIS passwords:

- Only the owner or superuser can change a NIS password. The superuser must know the current NIS password to change another user's NIS password.
- Only the first eight characters of the NIS password are significant; the rest are truncated.
- An NIS password must contain at least five characters if it includes a combination of *either one* of the following:
 - Uppercase and lowercase letters.
 - Alpha-numeric characters.
- An NIS password must contain at least four characters if it includes a combination of uppercase letters, lowercase letters, and numeric characters.
- An NIS password must contain at least six characters if it includes only monospace letters.
- You can change an NIS password in the NIS `passwd` map using `yppasswd` only if `rpc.yppasswdd` is running on the master server. (See `yppasswdd(1M)` in the *HP-UX Reference*.)

NIS Password

Refer to the following steps to create or change your NIS password in the NIS `passwd` map:

1. Execute the `yppasswd` command:

```
yppasswd user_login_name
```
2. The system prompts you for the old NIS password even if one does not exist. If it does exist, enter the old NIS password; otherwise, press [RETURN]. *Note:* The NIS password may be different from the one on your local node.
3. The system prompts you for the new NIS password twice to ensure you enter the correct response. Enter your new NIS password twice, pressing [RETURN] after each entry. The system now updates the master server `passwd` map.

Log Files

Using the `-l` option, you can execute `ypbind`, `ypserv`, and `yppasswdd` so that diagnostic and error messages are written to log files as shown in the following examples:

```
/etc/ypbind -l ypbind_log_file
/usr/etc/ypserv -l ypserv_log_file
/usr/etc/rpc.yppasswdd -l yppasswdd_log_file
```

Preceding each message logged to the file are the date, time, host name, process ID, and daemon name generating the message. Since the messages are uniquely identified by this information, these daemons can share a single log file.

If you execute the daemons without the `-l` option, the following responses occur:

- The `ypbind` daemon writes its messages directly to the system console, `/dev/console`.
- The `ypserv` daemon writes its messages to the `/usr/etc/yp/ypserv.log` file if it exists when `ypserv` is started.
- The `yppasswdd` daemon provides no messages.

The `ypxfr` command appends transfer information (which map from which server and how many entries it has) to the file `/usr/etc/yp/ypxfr.log` if it exists. The logging occurs only if `ypxfr` is not being run directly by someone at a terminal.

EXAMPLE: Logging occurs if the log file exists and `cron` is running `ypxfr` directly, using a crontab entry like the following one:

```
30 * * * * /usr/etc/yp/ypxfr nis_map
```

All log files could potentially grow without limit until they use up the available directory space. To avoid this occurrence, periodically check the file sizes. One method of preventing this problem is to create a crontab entry for each log file as follows:

```
0 1 * * 1,3,5 cat /dev/null > log_file
```

This line truncates `log_file` at 1:00 A.M. every Monday, Wednesday, and Friday.

Create Non-standard NIS Maps

You must be superuser to create and propagate non-standard NIS maps.

The `/usr/etc/yp/ypmake` file supports all of the standard maps shipped by HP. Non-standard maps are those maps which you create that are not originally supported by the `/usr/etc/yp/ypmake` file. To create them:

1. Modify `/usr/etc/yp/ypmake` on the master server so the map can be rebuilt. Modification requirements vary extensively. Generally, though, you need to filter a human-readable ASCII file through HP-UX utilities.

If the directory in which `/usr/etc/yp` exists supports only short file names (14 characters maximum), limit the new map name lengths to 10 characters maximum.

Note: However, the system automatically handles the longer standard NIS map names.

2. If using `Makefile` in `/usr/etc/yp` on the master server to build the maps, modify it so the new map can be rebuilt. (See `ypmake(1M)` in the *HP-UX Reference*.)
3. Modify `/usr/etc/yp/ypinit` on the master server to include the name of your new map in the list of `MASTER_MAPS`. Copy this modified script to all server nodes. This process ensures that any re-initialized or new slave servers will serve the new map.
4. For a client to access the data in the new map, it must exist on each of the servers. Execute the newly modified `ypmake` on the master server to build and copy the map to the current slave servers.

`/usr/etc/yp/ypmake`

Slave server support for the propagation of new maps consists of adding `crontab` entries or adding new entries to one of the `ypxfr` shell scripts described in the Propagate NIS Maps section.

The following sections cover one example for creating non-standard NIS Maps. The sections of the example include:

- Initial example environment.
- Modify `ypmake`.
- Modify `makefile`.
- Modify `ypinit`.
- Maintain a current access map on each slave server.
- Check the map's contents.

Initial Example Environment

Keep a list of the login names and the host names of all nodes on which each user is allowed to login:

- The information is stored in `/usr/etc/access_list`.
- The custom NIS map you wish to build from this file is `access`.

The general form of the ASCII file `/usr/etc/access_list` is as follows:

```
login_name1 [ host_name1 [ host_name2 ... ] ]
```

```
login_name2 [ host_name1 [ host_name2 ... ] ]
```

```
.  
. .  
.
```

```
login_namen  
[ # comments ]
```

- Each user has only one line.
- After each login name are zero or more host names. The user can log into any of these hosts.
- You can use both comments with a `#` (pound sign) in column one and blank lines.

The following samples could be in `/usr/etc/access_list`:

```
carole    alpha    catfish  handel
gerbil    catfish
```

`# bigmak is a new hire who has not yet arrived`

```
bigmak
mr_jad    axesys   handel
daveysan satie    yogurt
chum      handel   handel
speedy    handel   satie    catfish
fielding  alpha    beta     catfish
```

All of the users except for `bigmak` can log in on one or more systems.

You may want to use the login name as the key for storing this data in the access map so you can search the map with commands like `ypmatch`.

```
% ypmatch chum gerbil bigmak carole access
```

In the previous example, `ypmatch` command would provide an output like the following:

```
chum      handel
gerbil    catfish
bigmak
carole    alpha    catfish  handel
```

Modify ypmake

Modify `/usr/etc/yp/ypmake` on the master server as follows.

1. Insert a new function called `access()` after the `services()` function:

```
access() {
    grep -v "^[ ]*#" $1 | grep -v "^[ ]*$" | \
    awk 'BEGIN { OFS="\t"; } { print $1, $0 }' | \
    $MAKEDBM - $MAPDIR/access}
```

This function creates a map that has a key as the first field of each input record, creates a value that is the entire record, and skips over comment lines.

1. Add a new pattern to the case statement that is preceded by `for ARG in $*`; do.

You *must* place this information before the pattern `"*")` in the case statement:

```

access)
    if [ 'expr "$MAPS" : ".* access.*" -eq 0 ]; then
        MAPS="$MAPS access"
    fi;;

```

1. Add the new map name to the default list of MAPS to build. This addition ensures all maps are built (including the access map) if ypmake is called with no maps specified:

```

MAPS=${MAPS:-'passwd group hosts networks rpc \
services protocols netgroup access'}

```

1. Add a new pattern to the case statement that is preceded by for MAP in \$MAPS; do:

```

access)    build /usr/etc/access_list access;;

```

Modify Makefile

If using the makefile in /usr/etc/yp on the master server to build the maps, modify it as follows:

1. Insert a new variable called ACCESS after the SERVICES variable:

```

SERVICES = services services.byname
ACCESS = access

```

2. Add the new ACCESS variable to the definition of the ALL_MAPS variable:

```

ALL_MAPS = ${PASSWD} ${GROUP} ${HOSTS} ${NETWORKS} ${RPC}
${SERVICES} \
    ${PROTOCOLS} ${NETGROUP} ${ACCESS}

```

Modify ypinit

1. Modify the /usr/etc/yp/ypinit shell script on the master server to include the new map in list of all maps built on the master server:

```

MASTER_MAPS="group.bygid group.byname \
hosts.byaddr hosts.byname netgroup netgroup.byhost \
netgroup.byuser networks.byaddr networks.byname \
passwd.byname passwd.byuid protocols.byname \
protocols.bynumber rpc.bynumber services.byname \
access"

```

2. Copy this modified script to all current and future NIS servers.

Maintain a Current Access Map on Each Slave Server

1. Execute the newly modified `yppassd` on the master server to build and copy the access map to the current slave servers:

```
/usr/etc/yp/yppassd
```

2. On each slave server, modify the appropriate `yppassd` script to periodically copy the access map from the master server:

```
# yppassd_1perday - Perform daily NIS map check and
#updates
/usr/etc/yp/yppassd group.bygid
/usr/etc/yp/yppassd group.byname
.
.
.
/usr/etc/yp/yppassd access
```

Check the Map's Contents

Execute a few NIS commands to verify the success of your work:

```
% yppassd -m
services.byname    host1
.
.
.
access            host1
```

This `yppassd -m` command shows that the server you are bound to now serves the access map.

The order of the ypcat listing does not match the order of your file contents:

```
% ypcat access
fielding      alpha      beta      catfish
daveysan     satie     yogurt
speedy       handel    satie     catfish
mr_jad       xesys     handel
gerbil       catfish
carole       alpha     catfish   handel
bigmak
chum         handel
```

The following ypmatch command shows how you can selectively retrieve information from your new access map:

```
% ypmatch speedy daveysan fielding mr_jad access
speedy      handel    satie     catfish
daveysan    satie     yogurt
fielding    alpha     beta      catfish
mr_jad      axesys    handel
```


VHE Configuration and Maintenance

Virtual Home Environment (VHE) is an HP-developed service that allows you to configure login environments on remote nodes to mirror the login environment on the users' home nodes. VHE is available to any HP-UX system on a network running the NFS Services product.

You can choose whether to configure and use the service, although when you install NFS Services, VHE is also installed. For an overview of how VHE works, refer to the "NFS Services Overview" chapter.

Note The Network Information Service (NIS) is not mandatory for using VHE, but this chapter shows how to use VHE assuming NIS is configured and used.

If you do not plan to use NIS, you must have an alternate process for maintaining consistency of the `/etc/passwd` and `/etc/vhe_list` files for all nodes in the VHE group.

Configuration Overview

The following list is an overview of the steps you must complete to configure the nodes on your network with VHE. The steps are described in more detail after the overview list:

1. Prepare for configuring nodes with VHE by obtaining host names for the nodes in your network that will use VHE, installing and configuring NFS Services, and installing and configuring NIS (or instituting an alternate mechanism for maintaining consistent user and group IDs, internet address to host name mappings, password entries and `vhe_list`).
2. Compare VHE files in `/etc/newconfig` directory with existing files in the `/usr/etc/vhe` directory.
3. For each node, decide which directories or files are to be mounted and determine the names of mount point directories.
4. Create `/etc/vhe_list` on the NIS master server using the information from step 3.
5. Edit the `/etc/passwd` file on the NIS master server node to contain users' home directories which, in turn, contain the appropriate mount point directories.
6. Distribute the new `/etc/vhe_list` and `/etc/passwd` files by executing `ypmake` on the NIS master server.
7. On each node, edit `/etc/exports`.
8. On each node using VHE, execute `/usr/etc/vhe/vhe_mounter`.
9. Verify that VHE is running correctly.

Note You must be superuser to configure VHE.

1. Complete Preparation Steps

For each node that will use VHE, perform the following steps:

- Obtain a host name.
- Install and configure NFS Services.
- Install and configure NIS (or institute your own mechanism for maintaining consistent host names, group and password entries).

To obtain the host names for the nodes on your network that will use VHE, check your `/etc/hosts` file. If NIS is running, you can use the `ypcat hosts` command to look at the host information. If the BIND Name Server is configured, see the “Configuring and Maintaining the BIND Name Server” in the *Installing and Administering ARPA Services* manual.

To install and configure NFS Services, refer to the “NFS Configuration and Maintenance” chapter.

To install and configure NIS, refer to the “NIS Configuration and Maintenance” chapter. VHE can use NIS for file administration. For VHE to function, it needs all of the nodes using VHE to have a consistent view of the `/etc/passwd` and `/etc/vhe_list` files. NIS provides this; if you are not using NIS, you must ensure consistency by some other method.

The `/etc/vhe_list` file contains a list of all of the nodes that are using NFS to do the same remote mounts. (This is explained in detail in “4. Create `/etc/vhe_list`.”)

NIS maintains single versions of the `/etc/passwd` and `/etc/vhe_list` files on the NIS master server. From the NIS master server, you can add or delete users, change users’ home nodes and directories, and add or delete nodes from the VHE group. Once changes are made to `/etc/passwd` and `/etc/vhe_list`, the changes are made in the NIS maps and propagated to the NIS slave servers through the `ypmake` program.

2. Compare /etc/newconfig Files to Existing Files

When you installed the NFS services software, several new files were copied into the /etc/newconfig directory. Perform the following steps to prepare to configure VHE.

- Compare each /etc/newconfig file listed below with its counterpart shown in the following table.

File in /etc/newconfig directory	Counterpart in /usr/etc/vhe directory
vhe_mounter	vhe_mounter
vhe_script	vhe_script

- If the files are the same, skip to the next section, “Determine File Systems and Mount Point Directories.”
- If you have previously customized the files that exist in the /usr/etc/vhe directory, they will differ from those in /etc/newconfig. If there are differences, copy the current files in /usr/etc/vhe to a safe location and do *one* of the following:

Change the versions in /usr/etc/vhe to reflect the differences in the files in /etc/newconfig.

Copy the files in /etc/newconfig to /usr/etc/vhe. Then re-customize the newly copied files in /usr/etc/vhe if necessary.

3. Determine File Systems and Mount Point Directories

For each node that is using VHE, determine and write down the directories or files you want to mount and the directories you want to use as mount points. Use the following conventions when completing this step:

- Begin each mount point directory with a common path component. (In the examples for this manual, /vhe is used.)
- Attach to the above directory the host name of the machine you plan to mount. For example, for a machine named vic, the mount point directory is /vhe/vic. The machine name must match *exactly* the name returned by the hostname command (e.g., letters that are in lower case must be typed as lower case and letters that are upper case must be typed as upper case).
- For each directory that will be mounted from each machine to be connected with VHE, attach the directory name to the mount point name. To continue with the above example, if the machine vic has two directories or files to be mounted: / and /users, this would result in the directories for the two mount points to be /vhe/vic/ and /vhe/vic/users. In the case of

`/vhe/vic/`, you should delete the `/` at the end of the directory, resulting in the mount point `/vhe/vic`.

4. Create `/etc/vhe_list`

The `/etc/vhe_list` file contains a list of all directories that are mount points for your VHE environment. Each node accesses this list for the most current mount point information via NFS mounts. File systems of the remote node are mounted on the appropriate mount point using NFS.

To create the `/etc/vhe_list` file, complete the following items.

- As superuser, edit a file named `vhe_list` in the `/etc` directory of the NIS master server. The `vhe_list` file is installed at the time the NFS product is installed.
- For each mount point on each node create a one-line entry with the following form:

```
hostname file_system mount_point [mount_options]
```

Where:

hostname is the name of the node whose directory is mounted.

file_system is the name of the remote directory on the node to be mounted.

mount_point is the name of the local directory that acts as the mount point for the NFS mount.

mount_options is an optional field in `vhe_list` that contains options that are passed to the `mount` command. There should be no spaces between items in the *mount_options* field, and the items should be separated by commas. For example, to set the read and write size to 1024 bytes this field would look like:

```
rsize = 1024, wsize = 1024
```

Later, the `/usr/etc/vhe/vhe_mounter` script uses these fields to perform the appropriate NFS mounts. This script also creates the directories that will be the mount points, so it is not necessary for you to create these directories. If a file exists with the same name as one of the mount point directories, the script produces an error message. In this case, you need to either change the name of the existing file or change the name of the mount point directory.

If you are not using NIS, after you create the `/etc/vhe_list` file you need to distribute the `/etc/vhe_list` file to all the nodes in the VHE group.

Example: Simple Configuration with Single File System per Node

In the simplest case, each node has only one directory which is the root directory. Every node needs to have a set of directories for all members of the group. For example, consider a group consisting of the nodes A, B, C and D. A list of mount points for this group is `/vhe/A`, `/vhe/B`, `/vhe/C` and `/vhe/D`. Now taking these two lists, an `/etc/vhe_list` file with the following contents is created:

```
A / /vhe/A
B / /vhe/B
C / /vhe/C
D / /vhe/D
```

Example: Node with Multiple File Systems

Note If you do not have multiple directories or files on each node, you can go to “5. Update `/etc/passwd`.”

Doing mounts of several directories or files from one node requires some care in creating the `/etc/vhe_list` file. For example, if `/usr` is a separate directory on node C, and you execute the following on node A:

```
mount C:/ /vhe/C
```

An `ls` of `/vhe/C/usr` on node A shows it as an empty directory because NFS allows access to separate directories or files only if they are explicitly mounted.

This directory can be used to do a mount of the `/usr` directory of node C by executing the following on node A:

```
mount C:/usr /vhe/C/usr
```

Now an `ls` of `/vhe/C/usr` on node A shows the contents of the `/usr` directory on node C.

The example group is changed to show this complication with additional directories or files:

- A 1 directory under "/"
- B 2 directories or files one under "/"
and one under "/users"
- C 2 directories or files one under "/"
and one under "/usr"
- D 1 directory under "/"

When a node has multiple directories or files, you may choose to have all the directories or files mounted (as with C) or to have only some of the directories or files mounted (as with B). When `/usr/etc/vhe/vhe_mounter` is run, the mount point directories are created, if necessary, and the NFS mounts are made.

Using the rules outlined in "4. Create `/etc/vhe_list`," for the above group of nodes, you would create the following `/etc/vhe_list` file:

- A / /vhe/A
- B /users /vhe/B/users
- C / /vhe/C
- C /usr /vhe/C/usr
- D / /vhe/D

5. Update `/etc/passwd`

Update the `/etc/passwd` file on the NIS master server to force home directory access through the mount points. The entries in `/etc/passwd` should have the following form:

```
login_name:encrypted_password:UID:GID:comment:/vhe/hostname/home_dir:shell
```

Note If you are not using NIS, after updating the `/etc/passwd` file, you must distribute the changes to all nodes in the VHE group.

Example: `/etc/passwd` file entries before and after the VHE configuration

In this example, the first user's home directory is on node A; the second user's home directory is on node B; and the third user's home directory is on node C. All of the `/users` directories are in the root directories or files on their respective nodes.

Before VHE configuration:

```
andy::117:100:andy:/users/andy:/bin/csh
speedy::118:100:darren:/users/speedy:/bin/ksh
chum::119:200:Cris:/users/chum:/bin/sh
```

After VHE configuration:

```
andy::117:100:andy:/vhe/A/users/andy:/bin/csh
speedy::118:100:darren:/vhe/B/users/speedy:/bin/ksh
chum::119:200:Cris:/vhe/C/users/chum:/bin/sh
```

Example: Nodes with Multiple File Systems

Nodes with multiple directories or files do not change how the home directories are updated for VHE. For example, consider the following two entries in `/etc/passwd`. Fielding's home node is node B, which has two directories or files; Jeff's home node is node C, which has two directories or files. The nodes are from the example shown above.

Before VHE configuration:

```
Fielding::120:200:fielding:/users/fm:/bin/csh
Jeff::121:100:Jeff:/users/jbrl:/bin/csh
```

After VHE configuration:

```
Fielding::120:200:fielding:/vhe/B/users/fm:/bin/csh
Jeff::121:100:Jeff:/vhe/C/users/jbrl:/bin/csh
```

For node B, `/users` is its own directory and is mounted on the directory `/vhe/B/users`. This causes no change in the naming convention for the home directory. For node C, `/users` is on the root (`/`) directory. Node C also has another directory: `/usr`. If Jeff wants to be able to change the default directory to his mail file from `/usr/mail/jbrl` to `/vhe/C/usr/mail/jbrl` (to read mail via VHE), the `/usr` directory must be mounted on `/vhe/C/usr`.

6. Update `/etc/exports`

On each node that needs to export directories or files, edit the `/etc/exports` file to reflect all of the directories or files that are available for NFS mounting from each node. Details on this can be found in the "NFS Configuration and Maintenance" chapter.

7. Distribute /etc/vhe_list and /etc/passwd

To distribute the /etc/vhe_list and /etc/passwd files (i.e., make them accessible to all the nodes using NIS that are part of the same NIS domain), execute the following command on the NIS master server.

```
/usr/etc/yp/ypmake
```

This builds the NIS maps and propagates the maps to the NIS slave servers.

8. Execute /usr/etc/vhe/vhe_mounter

Note The /usr/etc/vhe/vhe_mounter script should be run when all nodes in the VHE group are powered up and ready for NFS mounting. If they are not ready for NFS mounting, then error messages are printed. These are not fatal errors; to recover from them you should retry vhe_mounter when the nodes are available for mounting.

The /usr/etc/vhe/vhe_mounter script uses the information in /etc/vhe_list to create the appropriate mount point directories on each node. When vhe_mounter notices that it is about to make a directory with the same name as the node from which vhe_mounter is executed, it makes a symbolic link with the same directory and links it to the node's root directory. When the vhe_mounter process completes running on each node, the proper mount points and symbolic links are created for each node.

The /usr/etc/vhe/vhe_mounter script also does NFS mounts using the appropriate directories to the remote machines on each node. When the mounts are complete, a node is ready for VHE.

To execute /usr/etc/vhe/vhe_mounter for each node separately, execute the following script on each node:

```
/usr/etc/vhe/vhe_mounter
```

To run /usr/etc/vhe/vhe_mounter for all nodes using VHE from a single node, execute the following as a batch file.

```
for i in `ypcat vhe_list | awk '{ print $1 }' | sort -u`  
do  
  remsh $i /usr/etc/vhe/vhe_mounter  
done
```

Note For this script to execute correctly, all nodes must be running ARPA/Berkeley Services with superuser capability allowed between the nodes when using remsh.

Example: This example shows the mount points and symbolic links resulting from the following `/etc/vhe_list` file:

A / /vhe/A
B / /vhe/B
C / /vhe/C
D / /vhe/D

The listing below shows the mount points and symbolic links for each node after the `/usr/etc/vhe/vhe_mounter` script completes running on each node (symlink = / denotes a symbolic link to the root (/) directory):

Node	/vhe/A	/vhe/B	/vhe/C	/vhe/D
A	symlink = /	Directory	Directory	Directory
B	Directory	symlink = /	Directory	Directory
C	Directory	Directory	symlink = /	Directory
D	Directory	Directory	Directory	symlink = /

9. Verify that VHE is Correctly Configured

To check if VHE is configured correctly, pick a login name that had a mount point added to its home directory. After `/usr/etc/vhe/vhe_mounter` has been run on each node, go to each node and log in using that selected login name (with the appropriate password). If VHE is correctly configured, the logins are successfully completed, and you are always placed in the execution environment associated with the selected login name.

Note You have now completed configuring the VHE service. The following sections describe advanced usage or set-up problems you may encounter when using VHE.

If you are configuring VHE as part of the NFS Services configuration, execute `/etc/netnfsrc` to complete the configuration procedure.

Configuration Refinements

The configuration procedure presented in the previous sections addresses most configuration cases. However, you may wish to refine your VHE configuration. This section explains how to refine your VHE configuration to allow NFS mounts to be done in the background.

NFS mounts in the Background

You can alter the `/usr/etc/vhe/vhe_mounter` script to allow mounts to be done in the background. This eases the situation where all nodes are not ready to respond when a node tries to mount them. To mount nodes in the background, you need to edit the `/usr/etc/vhe/vhe_mounter` script.

The `vhe_mounter` file has a shell variable called `BACKGROUND_MOUNT` whose initial value is set to 0. To allow nodes to be mounted in the background:

- Use an editor to set the value to something other than 0.
- Save the file and execute the `/usr/etc/vhe/vhe_mounter` script.

These changes cause NFS mounts to occur in the background. If the mounts are not successful on the first try, the NFS mounts continue to execute in the background.

Note Because each mount executes as a separate process until it completes or until the `retries` option for the NFS mount is exceeded, there may be a problem if there are many nodes (more than 30) in the VHE group.

VHE Maintenance

To keep VHE running correctly and efficiently, refer to the following sections.

Unmounting directories or files

If needed, you can unmount all of the remotely mounted directories or files. The easiest method of doing this is to execute the following:

```
umount -a -t nfs
```

This command can only be used when there are no VHE users logged on. If VHE is currently being used, the mount point directories will be busy and `umount` will not unmount a directory that is busy.

Just as having multiple directories or files available for remote mounting required mounting to be done in a specific order, unmounting directories or files must be done in the proper order. The order is just the reverse from the order that the mounts were done. The `umount` command with the “-a -t” options does this automatically.

For example:

```
mount A:/vhe/A  
mount A:/usr/vhe/A/usr
```

```
umount /vhe/A/usr  
umount /vhe/A
```

Adding or Deleting VHE Nodes

You may need to add or delete nodes from the VHE configuration. To do this, you need to perform the following steps:

1. Update the `/etc/vhe_list` on the NIS master server by either removing directories or files that are no longer available (if a node is being deleted) or adding directories or files that you want to become available (if a node is being added). Refer to the section in this chapter called “3. Create `/etc/vhe_list`” for more information about how to do this.
2. Edit the `/etc/passwd` file to show the addition of mount points to the home directory directory. Refer to the section in this chapter called “5. Update `/etc/passwd`” for more information on how to do this. If you are removing directories or files, you need to edit this file to delete mount points from the home directory directory.
3. To distribute the `/etc/vhe_list` and `/etc/passwd` files to the NIS servers, execute the following command on the NIS master server:

```
/usr/etc/yp/ypmake
```

4. Then execute the following:

```
/usr/etc/vhe/vhe_mounter
```

The script uses the information found in `/etc/vhe_list` to decide which new directories or files to mount. The `/usr/etc/vhe/vhe_mounter` script does not attempt to unmount a node deleted from the group. `vhe_mounter` needs to be executed on all of the nodes in the group for all of the nodes to be updated.

Advanced Usage

Adding altlogin and mounter Logins

The two logins of altlogin and mounter can be added to `/etc/passwd` by the superuser. This allows the user to:

- Log in using the mounter ID to complete NFS mounts to a node, if for some reason a node was not mounted when `vhe_mounter` was executed.
- Log in using altlogin to access the node where they currently are. This is useful if their home node is down.

These logins are similar to `who` and `date` because they execute a program. Mounter executes `vhe_u_mnt`, and altlogin executes `vhe_altlog` as follows:

- The `vhe_u_mnt` program executed by the mounter login only attempts to mount a directory of a node that is found in the `/etc/vhe_list` file. This prevents users from performing mounts to arbitrary nodes. Users can only perform mounts that could have been done by `/usr/etc/vhe/vhe_mounter`. If the node name entered at the prompt is not found in `/etc/vhe_list`, then an error message is printed and the mounts are not completed.
- The `vhe_altlog` program executed by altlogin prompts for a login ID and then attempts to do a `su` using the provided login ID. The user is then prompted for a password by `su`. If the proper password is given, the user is logged in with the home directory of `/tmp`. (If a proper password is not given, the user is not allowed access to the system.) Once logged in, none of the user's execution environment is available, but he or she can use the system.

To make these logins valid, you need to add them to the `/etc/passwd` file. Do this by adding an entry for each login to the `/etc/passwd` file. These entries should be similar to the following:

```
mounter::6:1:::/usr/etc/vhe/vhe_u_mnt
altlogin::6:1::/tmp:/usr/etc/vhe/vhe_altlog
```

The values shown in the above lines in UID, GID and home directory can be replaced with other values. Also note no password is provided in the above lines, but passwords can be entered if desired. If passwords are entered, tell the users allowed to use those logins what the associated passwords are because they *must* provide them when logging in.

Mounter Example

In this example, `dave` attempts to log in from node B when his home node, node A, is not mounted on node B. The following sequence would occur:

```
login: dave
Password:
Unable to change directory to "/vhe/A/users/dave"
```

```
login: mounter
Password:
Enter the name of the node to mount: A
```

```
login: dave
Password:
<Dave gets logged in >
```

Altlogin Example

This section shows an example of using altlogin. Julia is currently working at node B. Her home node A is not up, but Julia can gain access to node B in the following way:

```
login: altlogin
Enter your login name: Julia
Password:
%
```

Julia is now logged in at node B.

\$HOME

If you are writing scripts that make reference to files in a home directory, those file names should be prefixed with \$HOME (for sh or ksh). For csh, file names should be prefixed with a ~ character. This allows a file to be accessed in a consistent manner even if the home directory directory changes.

\$ROOT

To make a distinction between system files (like the password file) for the local and the home nodes, the following can be added to the .profile or .login file (home_node should be replaced with the name of the node):

```
ROOT = /vhe/home_node
export ROOT
```

This allows easier access to system files on a user's home node. For example, instead of typing:

```
more /vhe/home_node/etc/passwd
```


The user types:

```
more $ROOT/etc/passwd
```

Alternate Mount Points

The mount examples in this chapter are prefixed with `/vhe`. In addition to `/vhe` mount points, there may be other directories or files users in a VHE group want to regularly access.

For example, in a given VHE group, node A has directory `/Design`. To have a consistent view of this directory among all users in the VHE group, the `/Design` directory can be mounted on a directory `/Design`. To do this, the following line would be added to the `vhe_list` file:

```
A /Design /Design
```

Using VHE for Mail

To extend VHE to handle mail tasks:

- Change your default mailbox directory to have a mount point added to the beginning of it (just as the home directories are changed in `/etc/passwd`).
- Specify the above directory as the file to be used by the mail handler of your choice. If that mail file is on a separate directory, it must also be mounted to be available.

For example, if user `fm`'s home node is A, this shows how the `mailx` program can be invoked to read mail over NFS:

```
mailx -f /vhe/A/usr/mail/fm
```

In this example, if `/usr` was a separate directory on A, then the following would be added to `/etc/vhe_list`:

```
A /usr /vhe/A/usr
```

The NFS Automounter

This chapter explains how to mount and unmount file hierarchies automatically using automount.

Automount is an NFS tool that dynamically mounts and unmounts files and directories as needed.

For example, a user enters the following command:

```
vi /net/nodex/myfile
```

If the directory `/net/nodex` has been configured as an automount NFS directory, automount mounts `/net/nodex` for the user. Automount unmounts `/net/nodex` when it is no longer being used.

Automount provides the following features:

- Integration with NIS.

The automounter uses configuration information stored in maps that you can administer via NIS. This allows you to update file server information from the NIS master. You do not have to edit `/etc/checklist` files on every client when you add, modify or delete a server.

- Reduced possibility that a client hangs when a server crashes or is not available.

There is less dependency on the server because network file systems are mounted only when they are actually in use and unmounted when they are no longer in use.

- Replicated servers.

This allows you to specify a set of servers for a mount. The mount is done from the first server that responds. This allows a mount to succeed as long as at least one server in the set is available and provides some load balancing among the servers.

- A hosts map option that allows you to easily configure automount access for all exported file systems on the network.

Automount Concepts

Before configuring automount you must understand the following concepts.

When started, automount consults a series of maps for a list of directories or files to mount. To the kernel, it appears that a remote NFS file system has been mounted at each mount point. However, when the kernel thinks that it is sending NFS requests to the remote NFS server, it is actually sending them to a local automount daemon. The automount daemon then mounts the remote file hierarchy under `/tmp_mnt`. Finally, the automount daemon creates a symbolic link from the mount point specified in the map to the actual mount point under `/tmp_mnt`.

The hierarchy remains mounted for as long as it is needed. If a certain amount of time elapses without the hierarchy being used (the default is five minutes), automount unmounts it.

Automount Maps

Automount maps are used on clients to specify mount information. The mount information includes:

- the server's name
- file system pathname on the server
- local mount point
- mount options

There are three types of automount maps: master, indirect and direct.

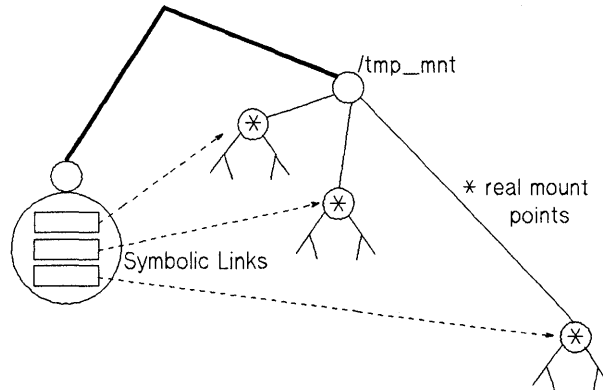
The master map lists all indirect and direct maps. You can also pass the indirect and direct map information on the command line.

An indirect map uses a common mount directory to make several mount points appear under a common directory.

A direct map contains mount information for any number of unrelated mount points. There is no common mount directory.

Indirect Maps

Indirect maps allow you to mount several file systems or directories from different sources and have them appear as subdirectories under a common mount directory. The automounter controls this common mount directory. The mount directory emulates a directory of symbolic links where each symbolic link points to the actual mount point under `/tmp_mnt`.



9-1. Indirect Map: Directory of Symbolic Links

An indirect map is useful when you have mounted directories from different sources that you want to appear under a common directory. For example, you may have engineers on different systems working on document files. You could import files from several different systems and have them all appear under the directory `/doc`.

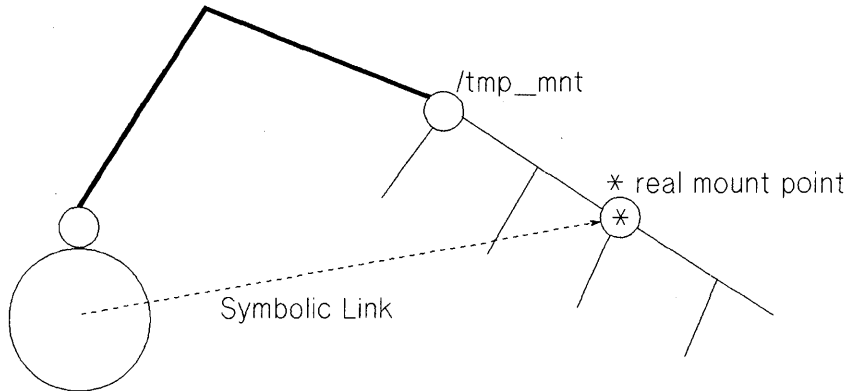
Thus, you could have `/doc/drivers`, which is imported from systemA, and `/doc/tools`, which is imported from systemB.

Because automount creates and controls the mount directory, no user can create subdirectories directly below `/doc`. If the directory `/doc` already exists, automount hides it and users are unable to access anything that had been below it. If you want a directory with a mix of automount mount points and local subdirectories you must use a direct map.

Also note that only the currently mounted file hierarchies appear in the mount directory. That is, an `ls` of the mount directory shows no subdirectories if no file hierarchies are currently mounted. Although this is confusing, this keeps automount from having to mount all the remote file hierarchies when a user executes an `ls` or `find` command on the mount directory.

Direct Maps

A direct map contains mappings for any number of unrelated mount points. There is no common mount directory. Each entry is a symbolic link to the actual mount point under `/tmp_mnt`.



9-2. Direct Map: Symbolic Link

For example, you may want to automount `/usr/man` and `/usr/lib`. It is inappropriate to use `/usr` as an indirect mount point, since there are many other subdirectories under `/usr`, such as `/usr/etc` and `/usr/bin`, that you would not want to hide. Instead, you should configure `/usr/man` and `/usr/lib` as two separate automount points in a direct map (You could also configure each one in a separate direct map).

Direct mount points are always visible. This means that when a directory that contains direct mount points is read, automount must mount any unmounted mount points. Although this behavior is less confusing for users, direct maps usually generate more mount requests than indirect maps.

Hosts Map

The hosts map is a quick way to specify automounts of all exported file systems in the network. The hosts map is a special form of an indirect map where the remote system name is used as the subdirectory name under the mount point.

For example, `/net` is typically used as a hosts mount point. If the system has been configured as such, a reference to `/net/systemB/users` would initiate an automatic mount of all file systems

from systemB that systemB has exported to this system. References to any directory under /net/systemB would refer to the corresponding directory on systemB.

Since the hosts map is a special form of an indirect map, the mount directory (/net) is controlled by automount and should not already exist.

Note that with a hosts map, the exported file systems of all the nodes in the hosts database (/etc/hosts, NIS hosts map or Domain Name Server database) are available. A user may then inadvertently cause a mount of a file system over X.25 or SLIP, which is unsupported, or through a slow router or gateway. Mounts over slow links may cause excessive retransmissions and degrade performance for all users.

Also note that a hosts map causes automount to mount all file systems that the server in question exports. The exported path names are sorted by length and mounted in order, shortest name first. Periodically, automount tries to unmount idle file systems from the bottom up (in the reverse order that they were mounted). If one of the directories at the top is still busy, automount stops unmounting and must remount the file systems it just unmounted.

Automount Configuration Checklist

The basic configuration steps are listed below and detailed in the the sections that follow.

1. Plan and design the implementation.
2. Configure the nodes for NFS if they are not already configured.
3. Create the master map file.
4. Create the direct and indirect map files.
5. If you will be using NIS to administer the maps, integrate the maps with NIS.
6. Edit `/etc/netnfsrc2` to add any command line options for starting automount.
7. Verify installation.

Planning and Design

Draw a network map of all file systems, their associated mount points and what systems they will mount. Note that if you are using the HP Cluster environment you do not have to run automount on the Cluster clients.

Select any NFS mount options you want to use.

Determine if you will use NIS to manage the maps or if you will use local files on each system.

Decide if you will create a master map or if you will specify the indirect and direct map names on the command line.

Decide what type(s) of mapping you will use (indirect, direct, hosts). Here are some points to consider when selecting map types:

The main difference between an indirect map and a direct map is that the indirect map uses a common mount directory that automount manages. The indirect map's mount directory cannot contain any local files or subdirectories.

Another difference is that direct map mount points are always visible, but indirect map's mount directories only show the mount points that are actually mounted. Although direct maps are less confusing for users, they usually generate more mount requests.

Configure for NFS

Configure the systems for NFS if you have not already done so. This task includes adding NFS to the kernel, editing `/etc/netnfsrc`. On the servers, edit `/etc/exports`.

Refer to the chapters, Installation and NFS Configuration and Maintenance, in this manual for details.

Create Master Map

The master map contains the names of the indirect and direct maps. It also contains the names of common mount directories for indirect maps.

You do not need a master map if you pass all the map information to automount in the command line.

The master map is usually created as `/etc/auto.master` and then made into the NIS map `auto.master`. By default, automount tries to get master map information from the NIS map `auto.master`. If a local master map file is specified on the command line, automount reads it before reading the NIS `auto.master` map.

The format for indirect map entries in the master maps is:

```
mount directory          indirect map [-mount options]
```

The format for direct map entries in the master map is:

```
/-                      direct map  [-mount options]
```

where:

- `mount directory` is the absolute path of the mount directory for the indirect map. Automount manages this directory. The directory should not be an existing directory, if a local directory by the same name already exists, automount will cover it.
- `/-` indicates that the entry is for a direct map.

- indirect map or direct map is the file name or NIS map name of the indirect map or direct map. If the map name is prefaced with a plus (+), automount searches for an NIS map. Refer to section, Integrating with NIS, in this chapter for more information.

This field can also be one of the following special maps:

- hosts Indicates the hosts map. The name of the remote host is used as the subdirectory name under the mount directory (Refer to the section, Hosts Map, earlier in this chapter for more information).
 - password Indicates the password map (Refer to the section, Configuring a Password Map, later in this chapter for more information.)
 - null Cancels a previous map for the indicated mount directory. For example, if you use this in a local master map, it cancels the entry in the NIS auto.master map.
- mount options can be any NFS mount options. Mount options in the master map are overridden by mount options in the indirect/direct maps.

Examples

The following master map references the indirect map `/etc/auto.doc` and the direct map `/etc/auto.direct`:

```
/doc                                /etc/auto.doc
/-                                  /etc/auto.direct
```

The following map would implement hosts mapping under the mount directory `/net`:

```
/net                                -hosts
```

Create Direct and Indirect Maps

Automount maps are usually named `auto.xxx`, where `xxx` is the name of the map. The name does not have to correspond to any mount points, but it is recommended that the map name correspond to the directory contents (for example, `auto.man` for man pages).

As with all NIS maps, names must be 10 characters or less if you have file systems that do not allow file names longer than 14 characters. This is because NIS adds four-character suffixes (`.dir` and `.pag`) to the map name.

By convention, maps are usually created under `/etc/autoconfig` or `/etc`. If you use NIS to administer the maps, you should create the source files under the `yppmake (1m)` source directory; the default `yppmake` source directory is `/etc`.

Indirect Maps

Each indirect map has a mount directory associated with it that is controlled by automount. The mount directory contains symbolic links to the actual mount points under `/tmp_mnt`. The mount directory is specified on the automount command line or in the master map.

The format of the indirect map entries is:

key [-mount options] server:directory

where:

- key is the name of the subdirectory under the mount directory.
- mount options can be any valid NFS mount options. This field is optional. Mount options specified here override any mount options specified in the master map.
- server:directory specifies the remote server name and the path of the remote file hierarchy (file system or directory) to mount.

Special Automount Characters

Automount recognizes special characters in direct and indirect maps to be used for substitutions and to escape other characters. They are:

& can substitute key values into the directory path names. For example:

```
jack -ro,intr server:/users/&
```

is the equivalent of:

```
jack -ro,intr server:/users/jack
```

***** is recognized as a catch all entry (a wildcard). It is the last or only entry in a map. It matches all keys and provides a value for the **&** substitutions that may exist in the right-hand side of a map. For example:

```
* -ro,intr server:/users/&
```

+ *mapname* The contents of another map can be included within the current map. If *mapname* is a directory with no slashes, automount interprets it as an NIS map. If the directory has slashes then automount looks for a local map with that name.

Example

The following indirect map, `/etc/auto.doc`, is used with the mount directory `/doc`:

```
drivers                -ro,soft    systemA:/doc/projectX/drivers
tools                  systemB:/tools/man
```

In this example, a reference to `/doc/tools` causes a mount of `/tools/man` from `systemB` if that hierarchy is not already mounted. The files mounted from `systemA` will not be edited by local users, so that hierarchy is mounted as a read-only soft mount.

Direct Maps

Direct maps can contain any number of unrelated mount points. No common mount directory is maintained.

The format of the direct map entries is:

```
key                               [-mount options]           server:directory
```

where:

- `key` is the absolute path of the mount point.
- `mount options` can be any valid NFS mount options. This field is optional. Mount options specified here override any mount options specified in the master map.
- `server:directory` specifies the remote server name and the path of the remote file hierarchy (file system or directory) to mount.

The characters in the earlier subsection, Special Automount Characters, are also valid for direct maps.

Example

The following direct map, `/etc/auto.direct`, contains entries for `/usr/man` and `/usr/lib`:

```
/usr/man           -ro,intr           systemD:/usr/man  
/usr/lib           -ro,intr           systemD:/usr/lib
```

Integrating Automount With NIS

Automount maps can be local files or administered as NIS maps.

By default, automount tries to read master map information from the NIS map `auto.master`. Automount also reads master map information from a local file if you specify one on the command line.

The master map can contain NIS map names for the indirect and direct maps instead of file names.

To specify an NIS map, preface the map name with a plus (+).

In the following master map, the indirect map for `/tools` is the NIS map `auto.toolfiles`:

```
/tools                + auto.toolfiles        -rw,intr
```

In addition, a local indirect or direct map file may include NIS maps. This allows you to have a local map file with system-dependent mappings and also use the NIS maps.

For example, your NIS domain may have the NIS map `auto.direct`, which is a direct map with entries for `/usr/man` and `/usr/lib`. You can add this to a local direct map file that contains an entry for `/usr/games`:

```
/usr/games            -ro,intr        systemB:/usr/games  
+ auto.direct
```

Creating NIS Maps

To create NIS maps, edit the `/usr/etc/yp/ypmake` shell script as shown below. For more information on using and configuring NIS, refer to the chapter, `NIS Configuration and Maintenance`, in this manual.

Remember that if you have file systems that do not allow file names longer than 14 characters the map names must be 10 characters or less. This is because NIS adds four-character suffixes (`.dir` and `.pag`) to the map name.

In the following text, assume that you want to create NIS maps for auto.master and a direct map called auto.xxxx. Within the scripts, auto_master and auto_direct are used for local variable names to avoid regular expression evaluation, but the map and file names are auto.master and auto.direct.

Under the function section, add the following blocks for auto.master and auto.xxxx:

```
auto_master() {
    grep -v "^[" ]*#" $1 | grep -v "^[" ]*$" | \
    awk 'BEGIN { OFS="\t"; } { print $1, $2, $3 }' | \
    $MAKEDBM - $MAPDIR/auto.master
}

auto_xxxx() {
    grep -v "^[" ]*#" $1 | grep -v "^[" ]*$" | \
    awk 'BEGIN { OFS="\t"; } { print $1, $2, $3 }' | \
    $MAKEDBM - $MAPDIR/auto.xxxx
}
```

In the block

```
for ARG in $*; do
    case "$ARG" in
```

add an entry for auto.master and auto.xxxx before the "*" in the case statement as follows:

```
auto_master )
    if [ 'expr "$MAPS" : ". * auto.master.*" -eq 0 ]; then
        MAPS="$MAPS auto.master"
    fi;;

auto_xxxx )
    if [ 'expr "$MAPS" : ". * auto.xxxx.*" -eq 0 ]; then
        MAPS="$MAPS auto.xxxx"
    fi;;
```


In the definition for

```
MAPS=${MAPS:-'passwd groups hosts ...
```

add an entry for auto_master and auto_xxxx.

In the block

```
for MAP in $MAPS; do
  case $MAP in
```

add an entry for auto.master and auto.xxxx as follows:

```
  auto_master)          build $DIR/auto.master auto.master ;;
  auto_xxxx)            build $DIR/auto.xxxx auto.xxxx;;
```

You should also modify /usr/etc/yp/ypinit on the master server. Add auto.master and auto.xxxx to the list of MASTER_MAPS.

Command Line Options

Automount is started by executing `/usr/etc/automount` at boot time. It is executed by default from the `/etc/netnfsrc2` file.

You can specify map information on the command line. Any map information you specify is read before information in the NIS map `auto.master` and takes precedence over the NIS map.

- m** ignores the initialization of maps within the NIS master map. This is useful for testing new indirect mount point and map combinations. A user can start automount without interference from entries in the master map.
- n** disables dynamic mounts. References to automount mount points succeed only when the target directory has been previously mounted. This is useful in preventing NFS servers from cross-mounting each other.
- T** turns on NFS call tracing and displays the expansion of NFS calls handled by automount on standard errors for debugging.
- v** logs event and status messages instead of just errors to `/usr/adm/syslog`. Refer to the `syslogd (1M)` man page for more information.
- D *envar = value*** assigns a value to the indicated variable within automount's environment.
- f *master-file*** reads a local master map file for initialization before the NIS master map. This allows each machine to have unique master map entries when the NIS master map exists. This option can be used when the NIS master map is empty or ignored with the `-m` option.
- M *mount-directory*** mounts temporary directories or files in a named directory other than `/tmp_mnt`.
- tl *duration*** is an absolute timer that will go off every *duration*, in seconds, that a directory is to remain mounted when not in use. The default is set for 300 seconds (five minutes).
- tm *interval*** specifies an interval, in seconds, of the amount of time between attempts to mount a directory. The default is 30 seconds.

<code>-tw interval</code>	specifies an interval, in seconds, of the amount of time between attempts to unmount directories or files that have exceeded their cached times. The default is 60 seconds.
<code>mountdir map mountopts</code>	specifies mount directory for indirect map and mount options. Refer to section, Create Master Map, in this chapter for more information.
<code>/- map mountopts</code>	specifies direct map and mount options. Refer to the section, Create Master Map, in this chapter for more information.

Examples

The following command tells automount to read the local file `/etc/auto.master` before reading the NIS `auto.master` map:

```
/usr/etc/automount -f /etc/auto.master
```

The following command tells automount to ignore the NIS `auto.master` map and use the local file `/etc/auto.master` only:

```
/usr/etc/automount -m -f /etc/auto.master
```

The following command tells automount to implement hosts mapping under the mount directory `/net`:

```
/usr/etc/automount /net -hosts
```

In the following example, the NIS map contains a hosts map entry under the mount directory `/net`. The local administrator does not want this system's users to have access to all the exported file systems of all the hosts in the network, so the administrator nullifies the `/net` entry in the NIS `auto.master` map:

```
/usr/etc/automount /net -null
```

The following command tells automount to mount items in the map `/etc/auto.myown` under the map directory `/doc`. Since automount reads this information first, any `/doc` entry in the NIS `auto.master` map is ignored:

```
/usr/etc/automount /doc /etc/auto.myown -rw,intr
```

In the following example, the file system to be mounted is on an optical disk device, so the amount of time that the file system remains mounted while idle is increased to ten minutes (600 seconds):

```
/usr/etc/automount -tl 600 /doc /etc/auto.myown
```

Verify Automount Configuration

To verify the automount configuration, try to access the mount points. Note that if you used an indirect map (including a hosts map) inactive mount points are not visible when the mount directory is read, so they are not listed with an `ls` command. You must access a mount point under an indirect mount directory (for example, with `cd` or accessing a file under the mount point) to test them.

If you are unable to access a mount point, try to manually mount the remote system in a temporary directory like `/tmp/test` with the `mount (1m)` command.

Check `/etc/mnttab` to see if an automount daemon has been started for your mount point. These entries have an entry of the form `server:(pid xxx)` for the first field. If you are using an NIS map, try running `automount` with local files.

Modifying the Automount Maps

Automount maps can be modified at any time. Automount may have to be restarted for some modifications to take effect. Rebooting the system is the safest and simplest way to restart automount.

Modifying the Master Map

Automount consults the master maps (NIS and/or local master maps) only at startup time. A modification to the master map will only take effect when automount is restarted.

Modifying Indirect Maps

Entries can be modified, deleted, or added to indirect maps without rebooting. The modifications take effect the next time the map is accessed.

Modifying Direct Maps

There are some update restrictions on direct maps. Modifications can be made to mount options or server names while automount is running. However, mount points can only change when automount is restarted.

Shutting Down Automount

Automount is normally started and stopped only when the machine is rebooted. To shut down automount gracefully during system operation, follow the following steps:

1. Make sure no processes have their current working directory set to any automount directories or subdirectories.
2. Then, send automount the SIGTERM signal (SIGTERM is the default signal sent by the kill command).

Automount cannot unmount file hierarchies or any indirect map's mount directory if any process is working in that directory (having been set with `cd`, `chdir`, or `fhdir`). Automount will loop and keep trying to unmount busy mount directories every ten seconds until it is able to unmount all the mount directories. The file hierarchies automount has mounted under `/tmp_mnt` only are unmounted after all the mount directories automount is serving have been successfully unmounted.

When the automount daemon first receives the SIGTERM signal, it forks a child automount process that will continue to service automount requests so processes that access mount directories served by automount will not hang. Two automount processes (the parent and child) will run until all mount directories served by automount (`/net` in the below example) have been successfully unmounted.

```
% ps -ef | fgrep automount | fgrep -v fgrep
  root 282  1 0 09:27:37 ?      0:00 /usr/etc/automount /net -hosts
% kill -SIGTERM 282
% ps -ef | fgrep autoomount | fgrep -v fgrep
  root 282  1 0 09:27:37 ?      0:00 /usr/etc/automount /net -hosts
  root 287 282 0 09:28:08?      0:00 /usr/etc/automount /net -hosts
%
```

If there are processes whose current working directory is `/net`, they need to have their current working directory changed. When the processes are no longer accessing any of the automount mount directories, the parent and child automount processes will finish cleaning up and exit. No further SIGTERM signals will be handled by these processes. Note: the background

processes can also have their current working directory set to a mount directory. This can prevent automount from unmounting that directory.

During the shutdown time, the child automount process will continue to service normal automount requests for any mount directories that have not yet been unmounted. The parent process will print the following message to /usr/adm/syslog once a minute until all processes accessing mount directories served by automount have terminated or are no longer accessing those mount directories:

```
Jul 21 09:45:08 hammer automount [298]: unmount failed - processes still accessing /net - will retry
```

If any processes are accessing the file hierarchies that were mounted on their behalf by automount (file systems mounted under /tmp_mnt) those hierarchies will still be mounted even after the automount processes terminate. For example, a user may have their current working directory set to /net/systemB as in the example above. Automount will successfully unmount the /net directory but not the systemB file hierarchy under /tmp_mnt. No other processes will be able to get to the systemB subdirectory from /net, however, systemB is still mounted under /tmp_mnt/net/systemB and is available with the umount command after automount terminates. For this manual unmount to succeed, no processes can have their current working directory set to any point within that file hierarchy. If automount is restarted and systemB is still mounted, automount will not automatically unmount it when not in use.

Warning **No other automount daemon should be started until the first has successfully cleaned up and exited. If a second automount daemon is started when the first is in its shutdown process, the second daemon will start its shutdown process. This means that there will now be four automount daemons: the first, the second, and their children. These daemons will not exit until all the mount directories they are serving have been unmounted**

Do not send the SIGTERM signal (kill -9, kill -KILL) to the automount daemon. This will cause any processes accessing mount directories served by automount to hang. The file hierarchies mounted by automount under /tmp_mnt will still be accessible.

Automount Error Messages

The following error messages are the most commonly seen if automount fails:

MESSAGE	bad entry in map <i>mapname</i> "key"
CAUSE	The entry "key" is invalid in the specified map.
ACTION	Correct the entry in the appropriate map.

MESSAGE	bad key <i>key</i> in direct map <i>mapname</i>
CAUSE	While scanning a direct map automount has found an entry key without a prepended "/".
ACTION	Keys in direct maps must be absolute pathnames.

MESSAGE	bad key <i>key</i> in indirect map <i>mapname</i>
CAUSE	While scanning an indirect map, automount has found an entry key containing a "/".
ACTION	Indirect map keys must be simple names, not path names.

MESSAGE	Cannot create socket for nfs: <i>rpc_err</i>
CAUSE	May indicate problems attempting to ping servers for a replicated directory. May be a network problem.
ACTION	Contact your network administrator

MESSAGE	Cannot create UDP service
CAUSE	Automount cannot establish a UDP connection
ACTION	Contact your network administrator.

MESSAGE	Can't get my address
CAUSE	Automount cannot find an entry for its host name through NIS, BIND, or /etc/hosts.
ACTION	Configure the system's hostname using NIS, BIND, or /etc/hosts.

MESSAGE **Can't mount *mount point*: reason**
CAUSE Automount could not mount its daemon at the mount point.
ACTION Make sure the specified mount point is a valid one.

MESSAGE **can't mount *server:directory*: reason**
CAUSE The mount daemon on the server refuses to allow automount to mount the specified file system.
ACTION Check the export table on the server.

MESSAGE **Can't update *directory***
CAUSE Automount was not able to update the mount table.
ACTION Check the permissions of the file, */etc/mnttab*.

MESSAGE **Couldn't create mount point *mount point*: reason**
CAUSE Automount was unable to create a mount point required for a mount. This usually occurs when attempting to hierarchically mount all of a server's exported directories or files.
ACTION A required mount point may exist only in a directory that cannot be mounted (it may not be exported) and it cannot be created because the exported parent file is exported read only.

MESSAGE **couldn't create *directory*: reason**
CAUSE Automount cannot create the directory */tmp_mnt* or the directory corresponding to the *-m* command line option.
ACTION Check the corresponding directory to see why it cannot be created.

MESSAGE ***dir mount point must start with '/'***
CAUSE Automount mount point must be given as absolute pathname.
ACTION Check the spelling and directory of the mount point.

MESSAGE	exiting
CAUSE	Automount has received a SIGTERM (has been killed) and is exiting.
ACTION	This is an advisory message.
<hr/>	
MESSAGE	hierarchical mount points: <i>pathname1</i> and <i>pathname2</i>
CAUSE	Automount does not allow its mount points to have a hierarchical relationship.
ACTION	Automount mount point must not be contained within another automounted directory.
<hr/>	
MESSAGE	host server not responding
CAUSE	Automount attempted to contact the server but received no response.
ACTION	Check to see if the server is up and running.
<hr/>	
MESSAGE	hostname: exports: <i>rpc_err</i>
CAUSE	Error getting export list from hostname.
ACTION	This indicates a server or network problem. Contact your network or system administrator.
<hr/>	
MESSAGE	leading space in map entry <i>entry text</i> in <i>mapname</i>
CAUSE	Automount has discovered an entry in an automount map that contains leading spaces. This is usually an indication of an improperly continued map entry.
ACTION	Correct the invalid map entry.
<hr/>	
MESSAGE	map <i>mapname</i>, key <i>key</i>: bad
CAUSE	The map entry is malformed and automount cannot interpret it.
ACTION	Recheck the entry.
<hr/>	

MESSAGE ***mapname: Not found***
CAUSE The required map cannot be located. This message is produced only when the -v option is given.
ACTION Check the spelling and the directory of the map name.

MESSAGE ***mapname: yp_err***
CAUSE Error in looking up an entry in an NIS map
ACTION May indicate NIS problems. Contact your system administrator.

MESSAGE ***Mount of server:directory on mount point: reason***
CAUSE Automount failed to do a mount.
ACTION This may indicate a server or network problem. Contact your network or system administrator.

MESSAGE ***mount point: Not a directory***
CAUSE Automount cannot mount itself on mount point because it is not a directory.
ACTION Check the spelling and directory of the mount point.

MESSAGE ***NFS server (pidnnn@mount point) not responding still trying***
CAUSE An NFS request made to the automount daemon with PID nnn serving mount point has timed out. Automount may be temporarily overloaded or dead.
ACTION Wait a few minutes to see if the condition persists. If so, exit all processes that use automounted directories, or change to a non automounted directory in the case of a shell, kill the current automount process and restart it again from the command line. If this does not work, reboot.

MESSAGE **nfscast: cannot receive reply:** *reason*
CAUSE Automount cannot receive replies from any of the servers in a list of replicated directories or files location.

ACTION Make sure the servers are up, running and reachable over the network

MESSAGE **nfscast: cannot send packet:** *reason*
CAUSE Automount cannot send a query packet to a server in a list of replicated directory locations.

ACTION Check to see if servers are up and running.

MESSAGE **nfscast:select:** *reason*
CAUSE Internal automount error

ACTION Contact your Hewlett Packard support contact.

MESSAGE **NIS bind failed**
CAUSE Automount was unable to communicate with the ypbind daemon. Please note: automount will continue to function correctly provided it requires no explicit NIS support.

ACTION If NIS is needed, check to see if there is a ypbind daemon running.

MESSAGE **no mount maps specified**
CAUSE Automount was invoked with no maps to serve and could not find the NIS auto.master map, it exited.

ACTION Recheck the command, or restart NIS.

MESSAGE **option ignored for key in mapname**
CAUSE Automount has detected an unknown mount option.

ACTION Correct the entry in the appropriate map.

MESSAGE **pathconf: server: server not responding**
CAUSE Automount is unable to contact the mount daemon on server that provides pathconf information.
ACTION Make sure the mount daemon is running on the server.

MESSAGE **pathok: couldn't find devid *device id***
CAUSE An internal automount error.
ACTION Contact your Hewlett-Packard support contact.

MESSAGE **remount server:directory on mount point: server not responding**
CAUSE Automount has failed to remount a file system it previously unmounted.
ACTION This message may appear at intervals until the directory is successfully mounted. No action is necessary.

MESSAGE **server:directory already mount on mount point**
CAUSE Automount is attempting to mount over a previous mount of the same directory. This could happen if an entry appears both in */etc/checklist* and in an automount map (either by accident or because the output of mount -p was redirected to */etc/checklist*).
ACTION Delete one of the redundant entries.

MESSAGE **server:directory - linkname : dangerous symbolic link**
CAUSE Automount is attempting to use server:directory as a mount point but it is a symbolic link that resolves to a directory referencing a mount point outside of */tmp_mnt* (or the mount point set with the -M option). Automount refused to do this mount because it could cause problems in the host's directory.
ACTION Check mount point name.

MESSAGE **server:directory no longer mounted**
CAUSE Automount is acknowledging that *server:directory* which was mounted earlier has been unmounted by the unmount command.
ACTION Automount will notice this within one minute of the unmount or immediately if it receives a SIGHUP. No action is necessary.

MESSAGE **svc_register failed**
CAUSE Automount cannot register itself as an NFS server.
ACTION Check to see if automount and portmapper are already running.

MESSAGE **trymany: servers not responding: reason**
CAUSE No server in a replicated list is responding.
ACTION This may indicate a network problem. Contact your network administrator.

MESSAGE **WARNING default option “option” ignored for map mapname**
CAUSE “*Option*” is an unrecognized default mount option for the map *mapname*.
ACTION Check and correct the entry in the map.

MESSAGE **WARNING: mount point already mounted on**
CAUSE Automount is attempting to mount over an existing mount point.
ACTION Check configuration on corresponding automount map.

MESSAGE **WARNING: mount point not empty!**
CAUSE The mount point is not an empty directory.
ACTION The directory mount point contains entries that will be hidden while automount is mounted there. This is only an advisory message.

MESSAGE **WARNING: *directory: line line number: bad entry***
CAUSE Automount has detected a malformed entry in the /etc/mnttab
file.
ACTION Check the corresponding entry in the /etc/mnttab file.

Advanced Automount Features

Automount can mount replicated directories or files from one of several potential servers. It can perform hierarchical mounts of all of a server's directories when any of them are referenced. Also, automount can provide better performance when the system administrator converts direct map entries into indirect maps and uses the automount subdirectory notation.

Replicated Servers

Automount provides a mount service for multiple locations with read only directories or files that are replicated across a network. A good example of this are man pages. In a large network, a set of manual pages may be available from several servers. Multiple mount locations can be listed in a map entry. For example:

```
/usr/lib    ro,intr    server1:/usr/lib    server2:/usr/lib
```

In this direct map example the man pages can be mounted from any of the servers. From this list of servers automount first selects the servers that are on the local network and pings these servers. If the list of servers contains non-local servers and no response is received from any of the local servers, it repeats the ping to all the servers in the list.

Once a server has been mounted, there is no status checking done. If the server goes down while the mount is in effect, the directory becomes unavailable. The only thing to do in this situation is to wait for the server to come up then try again. If the server goes down while the mount is not active and a user accesses the man pages, one of the other servers will be used.

Hierarchical Mounts

Automount can mount multiple directories from the same server in a hierarchy of mount points. Each directory is mounted on a subdirectory within another directory. The automount map syntax allows a hierarchy of mounts to be described. When the root of the hierarchy is referenced, automount mounts the whole hierarchy. The individual mounts within the hierarchy must all be NFS mounts, but they can be from different locations and have their own mount options. The mount points are relative to the mount root, not the host's directory root. There is no requirement that the first mount of a hierarchy be at the mount root. Automount will issue `mkdir`'s as needed to build a path to the first mount point if it is not at the mount root.

The symbolic link returned by automount to the kernel request is a path to the mount root. This is the root of the hierarchy that is mounted under `/tmp_mnt`. The mount point specification becomes important when mounting a hierarchy. Automount must have a mount point for each mount within the hierarchy:

```

/usr/local \
/      -ro,intr  bullwinkle:/usr/local      rocky:/usr/local \
/bin   -ro,intr  rocky:/usr/local/bin      bullwinkle:/usr/local/bin \
/man   -ro,intr  bullwinkle:/usr/local/bin/man  rocky:/usr/local/man


```

Note These mount point paths are relative to the mount root not the host's directory root. The first entry in this example has "/" as its mount point. It is mounted at the mount root.

Unmounts of the directories or files are done from the bottom up, in the reverse order of the mounts. If a higher-level mount point is busy then an unmount of the entire hierarchy fails. When automount fails to unmount a higher-level mount point, it must remount the directories or files it just unmounted. It walks back down the hierarchy from the busy mount point, mounting each directory. The remote server's directories or files are mounted on an all or nothing basis.

Subdirectory Notation

Remember that an indirect map contains entries with a key which is the name of the mount point/symbolic link in the directory, NFS mount options and the location:

morticia	-ro,intr	married:/adams1/morticia
gomez		married:/adams1/gomez
lurch		butler:/adams2/lurch
		
<i>key</i>	<i>mount options</i>	<i>location</i> <i>server:pathname</i>

Indirect maps with widely accessed Mount points may grow quite large and contain multiple references to a common remote directory. This sometimes causes duplicate mounts when there are multiple directories in the accessed remote directory.

Subdirectory notation can be used to prevent duplicate mounts from happening. It specifies the remote file system that should be mounted only if needed. If the directory is already mounted then automount makes a symbolic link to the new subdirectory of the common directory. The subdirectory notation is separated from the server:directory pair by a colon (:). For example, the following is a map for the directory `/users`:

```
dani          sharpei1:/users/sharpei1:dani
winston      sharpei1:/users/sharpei1:winston
```

When a user tries to access a file in `/users/dani`, automount mounts `sharpei1:/users/sharpei1`, but creates a symbolic link named `/users/dani` that points to the `dani` subdirectory in the temporarily mounted directory. Similarly, if a user immediately tries to access a file in `/users/winston`, automount immediately creates a symbolic link that points to the `winston` subdirectory. Notice that a second mount is not necessary.

Password Maps

A password map is a special form of an indirect map that uses the password database (`/etc/passwd` or the NIS `passwd` map) to locate a user's home directory for automount.

You enter the user's home directory in the password database in the form:

```
/dir/server_name/user_name
```

where:

- `/dir/server_name` together is the path where the user's home directory resides on the server. `server_name` is also the name of the server where the user's home directory resides.
- `user_name` is the user's name.

On the server, the user's home directory must exist as the same `/dir/server_name/user_name` specified in the password database.

The directory will be mounted as:

```
/mount_dir/user_name
```

where:

- `mount_dir` is the mount directory specified in the master map or on the automount command line. The mount directory `/home` is typically used with the password map.
- `user_name` is the user's name.

Since the actual mount point will not match the specification in the password database, you must create a symbolic link between the two:

```
ln -s /mount_dir/user_name /dir/server_name/user_name
```

Example

UserC's home directory files physically exist on the systemB, in the directory /autohome/systemB/userC

On the automount client, the /etc/passwd file has the following entry for userC, which lists /autohome/systemB/userC as the initial working (home) directory:

```
userC:prca5iX:210:200:bill user:/autohome/systemB/userC:/bin/ksh
```

On the automount client, you start automount as follows:

```
/usr/etc/automount /home -passwd
```

Since the password mapping will be mounted as /home/<username>, you must link the mount point so that it matches the entry in /etc/passwd:

```
ln -s /home/userC /autohome/systemB/userC
```


Common Commands

This chapter describes how to access files using NFS. It also explains how to use common NFS and Network Information Service (NIS) commands.

Note All references to *servers* and *clients* apply to NFS servers and clients unless preceded by NIS.

Key Terms

Term	Definition
Client	Can be defined in two ways: <ul style="list-style-type: none">- A node that requests data or services from other nodes (servers).- A process that requests other processes to perform operations. <p><i>Note:</i> An NFS client can also be configured as any combination of an NFS server, NIS client, or NIS server. (An NIS server must also be configured as an NIS client.)</p>
Cluster	One or more workstations linked together with a local area network (LAN), but consisting of only one root directory. For more information on cluster concepts, see <i>Managing Clusters of HP9000 Computers: Sharing the HP-UX Filing System</i> .
Cluster Auxiliary Server	A cluster client with a disk drive that contains files shared by the other members of the cluster.
Cluster Client	A node in an HP-UX cluster that uses networking capabilities to share directories or files, but does not have its root directory directly attached. For HP-UX 8.0, cluster clients can have locally mounted disks for local data storage.
Cluster Node (Cnode)	Any node operating in an HP-UX cluster environment, including cluster clients and cluster servers.
Cluster Root Server	The only node in an HP-UX cluster that has the root directory directly attached to it.
Context Dependent File (CDF)	A hidden directory that contains all the versions of a file needed by the different cnodes.
Export	To make a directory available to remote nodes via NFS.
File System	A directory structure used to organize files.
Host	A node that has primary functions other than switching data for the network.

Term	Definition
Internet Address	A four-byte quantity that is distinct from a link-level address and is the network address of a computer node. This address identifies both the specific network and the specific node on the network.
Key (NIS)	A string of characters (no imbedded blanks or tabs) that indexes the values within a map so the system can easily retrieve information. For example, in the passwd.byname map, the users' login names are the keys and the matching lines from /etc/passwd are the values.
Map (NIS)	A file consisting of logical records; a search key and related value form each record. NIS clients can request the value associated with any key within a map. NIS map is synonymous with NIS database.
Map Nickname (NIS)	A synonym for the NIS map name when using certain NIS commands.
Master Server (NIS)	The node on which one or more NIS maps are constructed from ASCII files. These maps are then copied to the NIS slave servers for the NIS clients to access.
Mount	To obtain access to a remote or local directory or directory (import).
Mount Point	The name of the directory on which a directory or part of a directory is mounted.
Network Information Service (NIS)	An optional network service composed of databases (maps) and processes that provide NIS clients access to the maps. The NIS service enables you to administer these databases from one node. NIS may or may not be active; check with your system administrator.
NIS Client	Can be defined in two ways: - A node that requests data or services from NIS servers. - An NIS process that requests other NIS processes to perform operations. <i>Note:</i> An NIS client can also be configured as any combination of an NIS server, NFS client, or NFS server. (An NIS server must also be configured as an NIS client.)
NIS Database	See Map (NIS).

Term	Definition
NIS Domain	A logical grouping of NIS maps (databases) stored in one location. NIS domains are specific to the NIS network service and are not associated with other network domains.
NIS Map	See Map (NIS).
NIS Password	<p>The password for a user's login ID that exists in the NIS passwd map. The password is the same one as the user password, but is administered through the NIS.</p> <p>You do not have to have a password to access the NIS databases.</p>
NIS Server	<p>Can be defined in two ways:</p> <ul style="list-style-type: none"> - A node that provides data (maps) or services to other nodes (NIS clients) on the network using NIS. - An NIS process that performs operations as requested by other NIS processes. <p><i>Note:</i> An NIS server must also be configured as an NIS client. It can also be configured as an NFS server, NFS client, or both.</p>
Node	A computer system that is attached to or is part of a computer network.
Server	<p>Can be defined in two ways:</p> <ul style="list-style-type: none"> - A node that provides data or services to other nodes (clients) on the network. - A process that performs operations as requested by other processes. <p><i>Note:</i> An NFS server can also be configured as any combination of an NFS client, NIS client, or NIS server. (An NIS server must also be configured as an NIS client.)</p>
Value (NIS)	A unit of information stored in NIS maps; each value has a corresponding key (index) so the system can easily retrieve it. For example, in the passwd.byname map, the users' login names are the keys and the matching lines from /etc/passwd are the values.

NFS Commands

This section explains how to access files via NFS and how to use the following NFS commands. The parenthetical comments refer to the *HP-UX Reference* sections where you can go for more information about these commands:

- *rpcinfo*(1M)
- *rup*(1)
- *rusers*(1)
- *showmount*(1M)
- *on*(1)

NFS Remote File Access

NFS allows many users to share the same files. Since access techniques are transparent, remote file access remains similar to local file access.

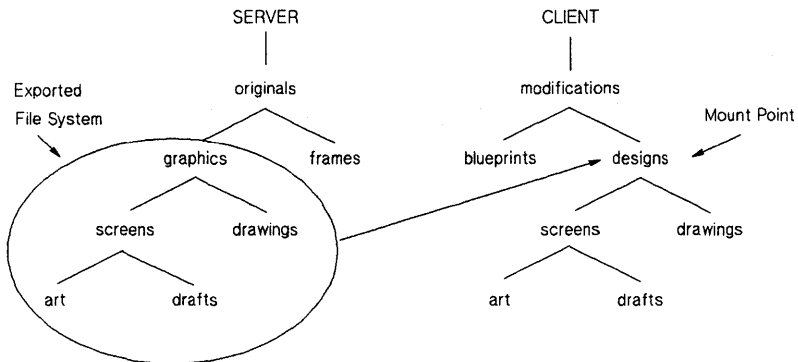
The superuser must perform two actions before you can access remote files via NFS:

- On the server, export the directory (i.e., make it available) to the client.
- On the client, mount (import) the directory.

Access to remote files is the same as for local files. You need to include either the complete path name starting with slash (/) or the path name relative to the current directory.

Note If operating in an HP-UX cluster environment and accessing a CDF (context dependent file) via an NFS mount, the CDF member is chosen based on the context of the NFS server, not the client node. Since this access method may return unexpected results, HP recommends you *do not* use CDFs with NFS.

EXAMPLE:



10-1. Using Commands with NFS Remote File Access

Example NFS Remote File Access	Example Entry
Edit the drafts file on the <i>server</i> .	server% vi /originals/graphics/screens/drafts
Edit the drafts file on the <i>client</i> .	client% vi /modifications/designs/screens/drafts
While in the frames directory, the <i>server</i> copies the art file from the screens directory.	server% cp /originals/graphics/screens/art art
While in the blueprints directory, the <i>client</i> copies the art file from the screens directory.	client% cp /modifications/designs/screens/art art
While in the screens directory, the <i>server</i> copies the art file to the frames directory.	server% cp art /originals/frames

Example NFS Remote File Access	Example Entry
While in the screens directory, the <i>client</i> copies the art file to the blueprints directory.	client% cp art /modifications/blueprints

rpcinfo

The `rpcinfo` utility verifies that the remote machine is capable of accepting and replying to an RPC request.

By providing a host name, you can list the registered RPC programs on a specific node. If you do not specify a host name, `rpcinfo` defaults to the local host.

There are command line options that can be used with the `rpcinfo` utility. They are:

`-p host` queries the portmapper about the remote *host* and prints a list of all registered RPC programs on that remote host. If no *host* is specified, it defaults to the local hostname.

For example, when you execute:

```
rpcinfo -p node_7
```

The system responds:

program	vers	proto	port	
100003	2	udp	2049	nfs
100004	2	udp	1028	ypserv
100004	2	tcp	1027	ypserv
100004	1	tcp	1027	ypserv
100007	2	tcp	1028	ypbind
100007	1	udp	1037	ypbind
100001	3	udp	1069	rstatd
100002	1	udp	1073	rusersd
100002	2	udp	1073	rusersd
100005	1	udp	1076	mountd
100008	1	udp	1078	walld
100012	1	udp	1080	sprayd

-u or -t. specifies what transport protocol (UDP or TDP) is used for a program on a specified host and report whether a response was received. The **-u** option tells `rpcinfo` to use the UDP protocol and the **-t** option tell `rpcinfo` to use the TDP protocol.

For example, when you execute:

```
rpcinfo -u node_2 mountd 1
```

The following system response indicates that the portmap daemon on `node_2` knows about program 100005 and that it is available:

```
program 100005 version 1 ready and waiting
```

- n *portnum*** use *portnum* as the port number for the **-t** and **-u** options instead of the port number given by the portmapper.
- b** Makes an RPC broadcast to the specified program and version number using UDP and reports all hosts that respond.
- d** deletes registration for the RPC service of the specified program and version number. Only users with the appropriate privileges can use this option.

rup

Execute `rup` to list host information, including how long they have been running, how many users are logged on to them, and their load average. By providing a host name, you can list information about a specific host.

When you execute `rup` specifying hostnames:

```
rup node_1 node_2 node_3 node_4
```

The last three columns of the system response show the load averages for 1, 5, and 15 minute intervals:

```
node_1    up           15:53,    load average: 0.11, 0.17, 0.15
node_2    up  2 days,    19:42,    load average: 0.00, 0.01, 0.01
node_3    up 21 days,    11:34,    load average: 1.66, 1.68, 1.60
node_4    up           19:24,    load average: 0.14, 0.18, 0.14
```

Executing `rup` without providing a hostname causes an RPC broadcast. The local node collects responses until the RPC times out (quits). This process generally takes about two minutes.

`rup` has three command line options that can be used:

- h** Sorts the display alphabetically by host.
- l** Sorts the display by load average.
- t** Sorts the display by up time.

For example, when you execute:

```
rup -t
```

The system responds:

```
collecting responses...
node_7      up 21 days,    11:28,    load average: 1.16, 1.42, 1.52
11.2.33.44  up 12 days,    22:15,    load average: 1.08, 0.82, 0.57
node_8      up 7 days,     18:27,    load average: 0.12, 0.09, 0.09
node_12     up 6 days,     21:20,    load average: 0.10, 0.08, 0.09
55.6.77.88  up 3 days,     3 mins,   load average: 0.00, 0.01, 0.01
node_6      up 2 days,     22:49,    load average: 0.00, 0.00, 0.02
99.0.11.22  up           18:14,    load average: 0.00, 0.00, 0.05
33.4.55.66  up           0 min     load average: 0.14, 0.04, 0.02
```

rusers

Execute **rusers** to list the host names and users logged in for all remote nodes. By providing a host name, you can list information about a specific remote node.

Executing **rusers** without providing a host name causes an RPC broadcast. The local node collects responses until the RPC times out (quits). This process generally takes about two minutes.

Example of Executing rusers

When you execute:

```
rusers
```

The system response displays the host name or internet address in the first column and the users in the second column:

```
77.8.99.00      root
node_6         user_4 user_3 user_8 user_11
node_3         u_2
node_1         u_7
11.2.33.44     root root
node_2         root u_5 root
node_16        test user
node_9         rootx root u_7 root
node_7         root
```

rusers has the following command line options:

- a** Gives a report for a machine even if no users are logged in on it.
- h** Sorts alphabetically by host name.
- i** Sorts by idle time
- l** You can list more extensive information by using the **-l** command: user, host, tty (terminal), login date and time, idle time (in minutes and seconds). In some cases, you can list the host that initiated the login session.

For example, when you execute:

```
users -l node_8 node_4
```

The third and fourth columns of the system response show the login date and time followed by the idle time:

```
rootx      node_8:console      Apr 07 14:00      20:29
user_3     node_4:ttyp03      Apr 12 08:09      :23 (node_5)
user_9     node_4:ttya00      Apr 08 09:24      10:42 (node_9:0.0)
```

In this example, the second line represents an rlogin or telnet session initiated from node_5. The third line represents an hpterm session displayed on the 0.0 display of node_9.

-u Sorts by number of users.

showmount

Execute `showmount` to list all the clients that have remotely mounted a directory. By providing a host name, you can specify the host. If you do not specify a host name, `showmount` defaults to the local host. For example, you might want to determine which nodes have your directories or files mounted.

`showmount` has the following command line options:

-a Prints all remote mounts in a client:directory format. The directory listed is the root of the directory that was mounted.

For example, when you execute:

```
showmount -a
```

The system response displays the client followed by the directory:

```
node_4:/tmp
node_7:/
node_2:/tmp
node_12:/usr/tmp/sys_rick
node_6:/tmp/y
node_8:/
```

-e Prints a list of exported directories or files.

For example, when you execute:

```
showmount -e node_7
```

The system responds:

```
export list for node_7:
/ node_31 node_32 node_1 node_6
/users/proj node_8 node_12
```

-d Lists directories that were remotely mounted by clients.

on

Use the `on` command to execute commands on a remote host. When executing the `on` command, you specify:

- A host on which to run the remote command.
- The command to run.
- Arguments for the command.

The `on` command then simulates your current environment on the server by passing your environment variables and information about your current working directory to the remote host. The `rexd` daemon on the server mounts the directory that contains your current working directory if it is not already mounted on the server. After the environment is simulated, the command executes in the simulated environment on the remote host.

Note Your environment is simulated on the remote host but not completely recreated. Execution of a given command on a remote host will not always produce the same results as the executing the command on your local computer. The simulated environment and the environment's limitations are discussed in the Environment Simulation in this manual.

The syntax of the `on` command is as follows:

```
on [-i | -n] [-d] host [ command [argument] ....]
```

Host specifies the name of the host on which to execute command. There must be an entry for `host` in the local computer's host data base.

Command specifies the command to execute on host. If `command` is not specified, `on` will start a shell on host.

You may specify three options (`-i`, `-n`, `-d`). The `-i` option must be used when invoking interactive commands, the `-n` option must be used when running commands in the background with job control, and the `-d` option is used when you wish to receive diagnostic messages.

Use of the `-d` option with either `-i` or `-n` is permitted. See the following examples:

```
on -i -d host
```

```
on -n -d host
```

You *cannot* use the `-i` and `-n` options at the same time.

-i The `-i` option invokes the interactive mode. This option must be specified for all interactive commands (commands which expect to be communicating with a terminal). Examples of interactive commands are `vi`, `csch`, and `more`. If this option is specified with a non-interactive command such as `sort`, it will be executed as an interactive command, but there may be no difference in behavior.

Example:

```
on -i node_7 vi file
```

-n The `-n` option sends the remote program an end-of-file when the program reads from standard input instead of connecting the standard input (`stdin`) of the `on` command to the standard input (`stdin`) of the remote command. The `-n` option is necessary when running commands in the background with job control.

-d The `-d` option allows you to receive diagnostic messages during the start up of the `on` command. The messages may be useful in detecting configuration problems if the `on` command is failing while connecting to a given host.

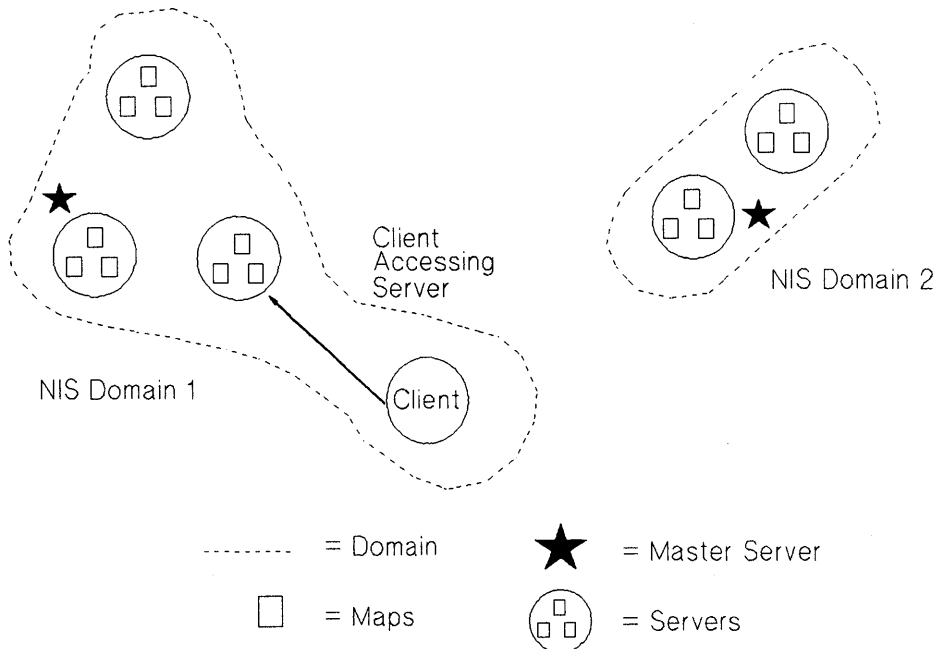
Network Information Service Overview

The Network Information Service (NIS) is an optional network database service that enables NIS clients to access information from any correctly configured NIS server on the network. (NIS was formerly known as Yellow Pages (YP), which is a registered trademark of British Telecommunications.) One NIS master server can automatically propagate modifications across the network.

NIS Maps

The NIS system stores information in NIS maps (databases) that are consistent across the nodes. Each map has a unique, case-sensitive map name that is used for accessing maps.

Each map consists of keys (for indexing) and values (data). You can use NIS commands for querying for values associated with a particular key within a map and for retrieving key-value pairs within a map.



10-2. Overview of NIS

NIS Servers and NIS Clients

NIS servers store and provide access to the NIS maps (databases); NIS clients request data from the maps residing on NIS servers. Since different NIS servers have consistent NIS maps, responses are identical no matter which NIS server answers a request. When doing NIS client and server configurations, keep the following information in mind:

- An NIS client can also be configured as any combination of an NIS server, NFS client, or NFS server.
- An NIS server must also be configured as an NIS client. It can also be configured as an NFS server, NFS client, or both.

NIS Domains

A NIS domain is a logical grouping of NIS maps; each NIS server contains a set of maps for at least one NIS domain.

NIS domains enable maps with the same names to exist on one LAN; the maps are made unique by belonging to separate NIS domains. With NIS domains, you need not worry about the maps interfering with each other because of the following:

- Each of the nodes within an NIS domain has the same NIS domain name.
- Maps using the same name in different NIS domains can have different contents.

NIS domains are implemented as subdirectories of `/usr/etc/yp` on the NIS servers only; the name of each subdirectory is the name of an NIS domain. For example, maps in the *research* NIS domain would be in `/usr/etc/yp/research`. (Note, NIS domain names are case sensitive.)

NIS Master and Slave Servers

Only two types of nodes have NIS databases: master servers and slave servers.

The NIS master server is the node on which all NIS maps within a particular NIS domain are created and modified. As modifications occur, the NIS slave servers copy the maps to ensure all NIS databases are alike; in turn, they provide resources to the NIS clients.

NIS clients can bind to either the NIS master server or to a slave server.

NIS Commands

Since NIS hides the details of how and where data is stored, you do not need to know all the configuration details to access information. You can, however, use the following commands to determine the location and content of NIS information. The parenthetical comment refers to the section in the *HP-UX Reference* where you can go for more information:

- `domainname(1)`
- `ypcat(1)`
- `ypmatch(1)`
- `yppasswd(1)`
- `ypwhich(1)`

domainname

Execute `domainname` to display the current NIS domain name:

```
domainname
```

For example, you might need to determine the current NIS domain name to define a `netgroup` in `/etc/netgroup`. (Netgroups are network-wide groups of nodes and users defined in `/etc/netgroup` on the master server.)

ypcat

Execute `ypcat` to list the contents of a specified NIS map. You can use either the map name or map nickname to specify the desired map.

Example of Executing `ypcat`

When you execute:

```
ypcat group.byname
```

(or)

```
ypcat group
```

The system response displays the group name, the group ID (GID), and the members of the group:

```
daemon::5:notes,anon,uucp
users::23>window,nowindow
other::1:root,daemon,uucp,who,date,dooley,sync
root::0:root
mail::6:root
sys::3:root,bin,sys,adm
rje::8:rje,shqer
bin::2:root,bin,daemon,lp
adm::4:root,adm,daemon
```

Example Using -x Option

To list the map nicknames applicable to the `ypcat` command, use the `-x` option.

When you execute:

```
ypcat -x
```

The system responds:

```
Use "passwd" for map "passwd.byname"
Use "group" for map "group.byname"
Use "networks" for map "networks.byaddr"
Use "hosts" for map "hosts.byaddr"
Use "protocols" for map "protocols.bynumber"
Use "services" for map "services.byname"
Use "aliases" for map "mail.aliases"
Use "ethers" for map "ethers.byname"
```

ypmatch

Execute `ypmatch` to print the data (values) associated with one or more keys in a specified NIS map. You can use either the map name or map nickname to specify the desired map.

To list the map nicknames applicable to the `ypmatch` command, use the `-x` option. For example, when you execute:

```
ypmatch my_node hosts.byname
```

The system response displays the internet address (value) associated with the `hosts.byname` map for the node `my_node`:

```
11.2.33.44 my_node
```


yppasswd

The NIS password is the password for a user's login ID that exists in the NIS passwd map. It is used as the user password, but is administered through NIS. Note, you are not required to have a password to access the NIS databases.

If you change your password with the `passwd` command, you will change only the entry in your local `/etc/passwd` file if the entry exists. If your password is not in the file, the following error message occurs when using `passwd`:

Permission denied.

If this error occurs, execute `yppasswd`.

NIS Password Guidelines

Execute `yppasswd` to change or install a password associated with a specified login name in the NIS passwd map.

The following list provides the requirements for creating and changing NIS passwords. *Note:* These guidelines are different from those of the `passwd` command (refer to the HP NFS Services vs. Local HP-UX appendix in this manual):

- Only the owner or superuser can change an NIS password. The superuser must know the current NIS password to change another user's NIS password.
- Only the first eight characters of the NIS password are significant; the rest are truncated.
- An NIS password must contain at least five characters if it includes a combination of *either one* of the following:
 - Uppercase and lowercase letters.
 - Numeric and special characters.
- An NIS password must contain at least four characters if it includes a combination of uppercase letters, lowercase letters, and numeric characters.
- An NIS password must contain at least six characters if it includes only monospace letters.

NIS Password

Follow these steps to create or change your NIS password in the NIS passwd map:

1. Execute the `yppasswd` command.

```
yppasswd user_login_name
```

2. The system prompts you for the old NIS password even if one does not exist. If it exists, enter the old NIS password; otherwise, press [RETURN].

Note The NIS password may be different from the one in your local `/etc/passwd` file.

3. The system prompts you for the new NIS password twice to ensure you enter the correct response. Enter your new NIS password twice, pressing [RETURN] after each entry. The system now updates the master server passwd map.

Example of Executing `yppasswd`

When you execute:

```
yppasswd
```

The system responds:

Old NIS password:

New password:

Retype new password:

The NIS passwd has been changed on host_name, the master NIS passwd server.

ypwhich

Execute `ypwhich` to print the host name of the NIS server supplying NIS services to an NIS client.

To list all available maps and their NIS master server host names, use the `-m` option. You can use either the map name or map nickname to determine which NIS server is the master server for a specified NIS map.

To list the map nicknames applicable to the `ypwhich` command, use the `-x` option. For example, when you execute:

```
ypwhich -m
```

The system response displays the available maps and their NIS master server host names:

services.byname	node_1
rpc.bynumber	node_1
protocols.bynumber	node_1
protocols.byname	node_1
passwd.byuid	node_1
passwd.byname	node_1
networks.byname	node_1
networks.byaddr	node_1
netgroup.byuser	node_1
netgroup.byhost	node_1
netgroup	node_1
hosts.byname	node_1
hosts.byaddr	node_1
group.byname	node_1
group.bygid	node_1
vhe_list	node_1
ypservers	node_1

Troubleshooting

If a node on the network is not operating correctly, use this chapter to identify and correct the problem. Most problems occur when:

- Installing the network.
- Changing the network (e.g., adding a node or extending the coaxial cable).
- Another system on your link, fails.

Before troubleshooting the problem, get or create your network map as described in the *Installing and Administering LAN*, *Installing and Administering FDDI/9000 Software*, or *Installing and Administering Token Ring/9000* manuals. Use this map when checking configuration and network layout information. Remember to update it any time you make a change to the network.

Note All references to servers and clients apply to NFS servers and clients unless otherwise specified.

Key Terms

Term	Definition
Client	<p>A node that requests data or services from other nodes (servers).</p> <p>A process that requests other processes to perform operations.</p> <p><i>Note: An NFS client can also be configured as any combination of an REX server, NIS client, or NIS server. (An NIS server must also be configured as an NIS client.)</i></p>
Cluster	<p>One or more workstations linked together with a local area network (LAN), but consisting of only one root directory. For more information on cluster concepts, see <i>Managing Clusters of HP9000 Computers: Sharing the HP-UX Filing System</i>.</p>
Cluster Auxiliary Server	<p>A cluster client with a disk drive that contains files shared by the other members of the cluster.</p>
Cluster Client	<p>A node in an HP-UX cluster that uses networking capabilities to share directories or files, but does not have its root directory directly attached. For HP-UX 8.0, cluster clients can have locally mounted disks for local data storage.</p>
Cluster Node (Cnode)	<p>Any node operating in an HP-UX cluster environment, including cluster clients and cluster servers.</p>
Cluster Root Server	<p>The only node in an HP-UX cluster that has the root directory directly attached to it.</p>
Daemon	<p>Background programs that are always running, waiting for a request to perform a task.</p>
Export	<p>To make a directory available to remote nodes via NFS.</p>
Hard Mount	<p>A mount that causes NFS to retry a remote directory request until it succeeds, you interrupt it (default option), or you reboot the system.</p>
Heterogeneous Cluster	<p>A diskless cluster with more than one type of computer attached.</p>
Homogenous Cluster	<p>A diskless cluster composed of nodes of only one type of computer architecture (e.g., HP 9000 Series 300)</p>
Host	<p>A node that has primary functions other than switching data for the network.</p>

Term	Definition
Map (NIS)	<p>A file consisting of logical records; a search key and related value form each record. NIS clients can request the value associated with any key within a map.</p> <p>NIS map is synonymous with NIS database.</p>
Master Server (NIS)	<p>The node on which one or more NIS maps are constructed from ASCII files. These maps are then copied to the NIS slave servers for the NIS clients to access.</p>
Mount	<p>To obtain access to a remote or local directory or directory (<i>import</i>).</p>
Mount Point	<p>The name of the directory on which a directory is mounted.</p>
Netgroup	<p>A network-wide group of nodes and users defined in <i>/etc/netgroup</i>.</p>
Network Information Service (NIS)	<p>An optional network service composed of databases (maps) and processes that provide NIS clients access to the maps. NIS enables you to administer these databases from one node.</p> <p>NIS may or may not be active; check with your system administrator.</p>
NIS Client	<p>A node that requests data or services from NIS servers.</p> <p>An NIS process that requests other NIS processes to perform operations.</p> <p><i>Note:</i> An NIS client can also be configured as any combination of an NIS server, NFS client, or NFS server. (An NIS server must also be configured as an NIS client.)</p>
NIS Database	<p>See Map (NIS).</p>
NIS Domain	<p>A logical grouping of NIS maps (databases) stored in one location. NIS domains are specific to NIS and are not associated with other network domains.</p>
NIS Map	<p>See Map (NIS).</p>
NIS Password	<p>The password for a user's login ID that exists in the NIS <i>passwd</i> map. The NIS password is the same one as the user password, but is administered through NIS.</p> <p>You do not have to have a password to access the NIS databases.</p>

Term	Definition
NIS Server	<p>A node that provides data (maps) or services to other nodes (NIS clients) on the network using NIS.</p> <p>An NIS process that performs operations as requested by other NIS processes.</p> <p><i>Note:</i> An NIS server must also be configured as an NIS client. It can also be configured as an NFS server, NFS client, or both.</p>
Node	<p>A computer system that is attached to or is part of a computer network.</p>
Server	<p>A node that provides data or services to other nodes (clients) on the network.</p> <p>A process that performs operations as requested by other processes.</p> <p><i>Note:</i> An NFS server can also be configured as any combination of an NFS client, NIS client, or NIS server. (An NIS server must also be configured as an NIS client.)</p>
Slave Server (NIS)	<p>A node that copies NIS maps from the NIS master server and then provides NIS clients access to these maps.</p>
Soft Mount	<p>An optional mount that causes access to remote directories or files to abort requests after one NFS attempt.</p>

Troubleshooting References

Troubleshooting the NFS Services primarily concerns the areas: power up and connectivity, NFS Services, NIS, VHE, and REX. This chapter only addresses NFS, NIS, VHE, and REX problems. Link diagnostics and troubleshooting are in the *Installing and Administering LAN* manual.

If your system is having problems communicating with or through a non-HP system, refer also to the appropriate user and system administration documentation for that system.

Power Up and Connectivity Testing

Refer to the following documentation if your system cannot communicate with other systems on the network:

- *LAN Interface Controller (LANIC) Installation and Reference Manual.*
- *HP Repeater Installation Manual* (only if you are using a HP 92223A repeater).
- *Installing and Administering LAN/9000.*
- *Installing and Administering FDDI/9000 Software.*
- *Installing and Administering Token Ring/9000.*
- *Installing and Updating HP-UX.*
- *EISA FDDI Adapter Installation Guide for the Series 700.*
- *HP-PB FDDI Adapter Installation Guide for the Series 800.*
- *EISA Token Ring Quick Installation for the Series 700.*
- *HP-PB Token Ring Quick Installation for the Series 800.*
- *HP-UX Reference.*
- *System Administration Tasks.*
- *LAN Cable and Accessories Manual.*
- *Installing and Administering Network Services.*
- *Installing and Administering ARPA Services.*

Troubleshooting Sections

Refer to the Troubleshooting NFS section or the *HP-UX Reference* if you cannot mount a remote directory, access a remotely mounted directory, or experience other problems with the NFS service.

Refer to the Troubleshooting NIS section or the *HP-UX Reference* if you configured the system to use the Network Information Service, but cannot access files serviced by it.

Refer to the Troubleshooting VHE section if you configured the system to use VHE, but it doesn't function as described in the VHE Configuration and Maintenance chapter.

Refer to the Troubleshooting REX section if you configured the system to use REX, but it doesn't function as described in the Remote Execution Facility (REX) chapter.

Guidelines

Troubleshooting is an elimination process that narrows a problem. If a process worked before but does not work now, first consider what has changed. For example, have you moved hardware or modified configuration files?

Start with the minimum number of variables, then gradually and selectively add other variables such as the following:

- If you cannot communicate with one system, try another one. If the second system works, the problem may be with the first remote system and not your system.
- If one system cannot communicate with yours, try another one. If neither system can communicate with yours but they can communicate with each other, the problem may be with your system.
- If one service does not work, try another one. The problem may be with a particular service to a particular system and not a problem with the system itself.

Common Network Problems

Network problems generally occur under the following circumstances:

- File permissions on the client or server restrict the operation.
- Network services on the client or server are misconfigured or malfunctioning.
- Network LAN, FDDI, or Token Ring software or hardware is misconfigured or malfunctioning.

Initial Troubleshooting

You should first check the following situations to ensure they are not the cause. If they are not, refer to the flowcharts in this chapter.

Configuration

1. Is your host running HP-UX 6.0 or later for the Series 300/400 and HP-UX 2.0 or later for the Series 600/700/800? For File Locking and REX, your host must be running HP-UX 6.5 or later for the Series 300/400 or HP-UX 7.0 or later for the Series 600/700/800. Execute `uname -a` or `uname -r` to check the HP-UX version number.
2. Does your system have the recommended 256K additional memory for networking software?
3. Is your HP 9000 a supported configuration? If you are unsure, contact your HP support representative.
4. Does the error occur on a node other than a Series 300/400 or Series 600/700/800? If so, refer to the appropriate system documentation.

Hardware

The *Installing and Administering LAN/9000* documentation contains details about troubleshooting hardware problems.

1. Are all connections along the network cabling tight?
2. Is each cable segment less than 500 meters for ThickLAN and less than 100 meters for ThinLAN?
3. Are there no more than two repeaters between you and the node with which you want to communicate?
4. Are you mixing Ethernet¹ hardware with IEEE 802.3² hardware? This is not an acceptable combination since they do not have the same electrical characteristics.
5. Is there a 50 ohm terminator at the end of each cable?
6. Is the MAU tapped correctly into the cable?
7. Is the cable grounded in only one place?
8. Is the AUI solidly connected to the interface card?

(1) Ethernet is a local area network system developed by Digital Equipment Corporation, Intel Corporation, and Xerox Corporation.

(2) IEEE 802.3 is a networking standard that is accepted by the Institute of Electrical and Electronic Engineers.

9. Is the host hardware working correctly?

Network Communication

1. Is the remote node HP certified? If you are unsure, contact your HP support representative.
2. Can any other two nodes on the network communicate? If not, the problem may be global. Refer to the *LAN Cable and Accessories Installation Manual* and *Installing and Administering LAN/9000*, *Installing and Administering FDDI/9000*, or *Installing and Administering Token Ring/9000* documentation.
3. Have you performed the corrective action supplied with the error message you received? Consult the appropriate entry in the *HP-UX Reference*.
4. If using gateways, do both hosts have routing information to each other? Refer to *route(1M)* in the *HP-UX Reference*.
5. If operating in an HP-UX cluster environment and trying to mount an NFS directory, ensure you are using the cluster server's host name (on which the directory is mounted) as the node specified in the mount command. This will be either the cluster root server or the cluster auxiliary server.
6. If operating in an HP-UX cluster environment and having link problems, cnodes will not be able to boot. Since link diagnostics reside on the root disk, first test the Link from the root server. (Refer to *Installing and Administering LAN/9000*, *Installing and Administering FDDI/9000* or *Installing and Administering Token Ring/9000* documentation.)

NIS and NFS Services

1. Is the client system trying to perform tasks as superuser on the remote system? Executing setuid root programs cannot access files or directories unless the permission other allows it.
2. Was network communication established between the client and server using the procedures outlined in the *System Administration Tasks* manual, and in the NFS Configuration and Maintenance and NIS Configuration and Maintenance chapters of this manual?
3. Is the problem associated with remote file locking? The lockf(2) call fails when attempting to lock a remote file. Prior to HP-UX release 6.5 for the Series 300/400 and HP-UX release 7.0 for the Series 600/700/800, NFS Services *did not* support file locking on remote directories or files.

4. Is the problem associated with attempts to access remote device files? Prior to release 6.5 for the Series 300/400 and HP-UX release 7.0 for the Series 600/700/800, HP-UX *did not* support remote access to device files.
5. Does the `inetd` security file (`/usr/adm/inetd.sec`) on the remote system limit access to the remote system for the RPC service you are trying to access?
6. Is the directory listed in the server's exports list and/or has the directory been exported via `exportfs` (Type `exportfs` to find out)?
7. Does `/etc/exports` restrict directory access to a specific netgroup or host?
 - a. The `/etc/netgroup` file must list the netgroup if it is specified in the exports list.
 - b. The `/etc/hosts` file must contain the host if it is specified either in the exports list or in `/etc/netgroup`.
8. Is the directory or file mounted? To check, execute the `mount` command.
9. If the directory is supposed to be mounted at boot time, is it listed in `/etc/checklist`?
10. If programs accessing remote files hang, is the NFS or NIS server down?
11. Is data on remote nodes corrupted? Ensure only one system is writing to the file at a time; NFS allows more than one client to write to a file simultaneously.

Remote Execution (REX)

1. Is the server configured to run `rex`d? The server must have an entry in `/etc/inetd.conf` in order to run `rex`d (see `rex`d(1M) in the *HP-UX Reference*).
2. Was network communication established between the REX client and the REX server using procedures outlined in the *System Administration Tasks* manual, and in the NFS Configuration and Maintenance, NIS Configuration and Maintenance, and Remote Execution Facility (REX) chapters of this manual?
3. Does the `inetd` security file (`/usr/adm/inetd.sec`) on the REX server limit access to the remote system for the `rex`d service?
4. Does the user have a user account on both the REX client and the REX server with matching UIDs?

5. Was `rexid` on the REX server started with the `-r` option? This causes access to be restricted based on `/etc/hosts.equiv` and the user's `.rhost` file on the REX server.
6. If the remote command is hung, is the NFS or NIS server down?
7. Is the problem associated with attempts to mount the directory containing the user's current working directory?
 - a. Is the directory in the NFS server's `/etc/exports` file?
 - b. Does the NFS server's `/etc/exports` entry for the directory restrict access to a specific `netgroup` or `host`?

Error Messages

The problem can exist on the server even though the error message may not occur on it.

Since most of the error messages are self-explanatory, you can determine the necessary corrective action when simple errors occur. For the other error messages, follow the corrective action supplied in the *HP-UX Reference* for that service. (These error messages are preceded by the name of the service.)

Stale File Error Messages

errnoA file can become "stale" is one NFS user removes that file when another NFS user still has that file open. For example, both user 1 and user 2 are sharing the directory and copy of code simultaneously:

user 1	user 2
\$ cd /source/pkg1	\$ cd /source
	\$ rm -rf pkg1

```
ls
: Stale File Error
```

To prevent this from happening, use a system that lets the users make individual copies of source files in their own directories as well as look to see how your users are using NFS directories and files.

Unsolved Problems

If you do not solve the problem after working through the previous troubleshooting steps and following flowcharts, call your HP support representative for assistance. Provide as much information about the problem as possible, including information from your network map and the following items:

- The activity you were attempting when the error occurred. Describe the HP-UX commands, job streams, result codes, and events leading to and including the problem.
- The version or update information for all software you are running. You should be able to find this information on your *install* or *update* media.
- The error messages you received. Record all error messages and numbers that appeared both on all nodes.
- The troubleshooting steps you tried.
- The problems you ruled out and why.

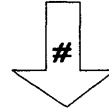
Flowchart Format

Each of the following flowcharts have a corresponding set of labelled explanations. You can use the flowcharts alone or with the explanatory text for more detail.

Start of Flowchart #



Go to and enter
specified Flowchart#



Make a decision



Perform an action



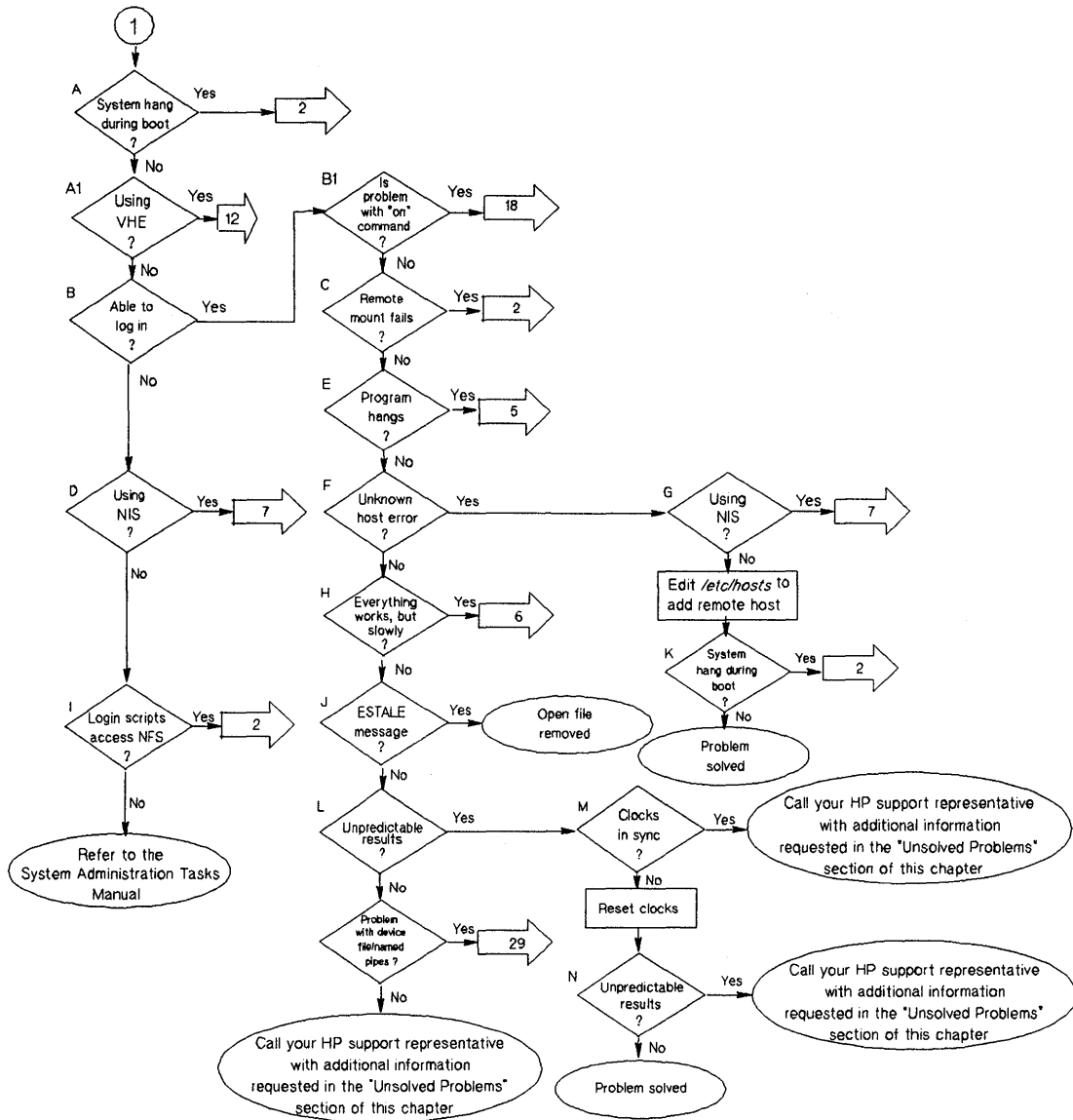
Exit Flowchart



Flowchart Symbols

Note These flowcharts are for HP systems. Processes referenced in the flowcharts may not be part of NFS products from other vendors (e.g., portmap).

Troubleshooting NFS



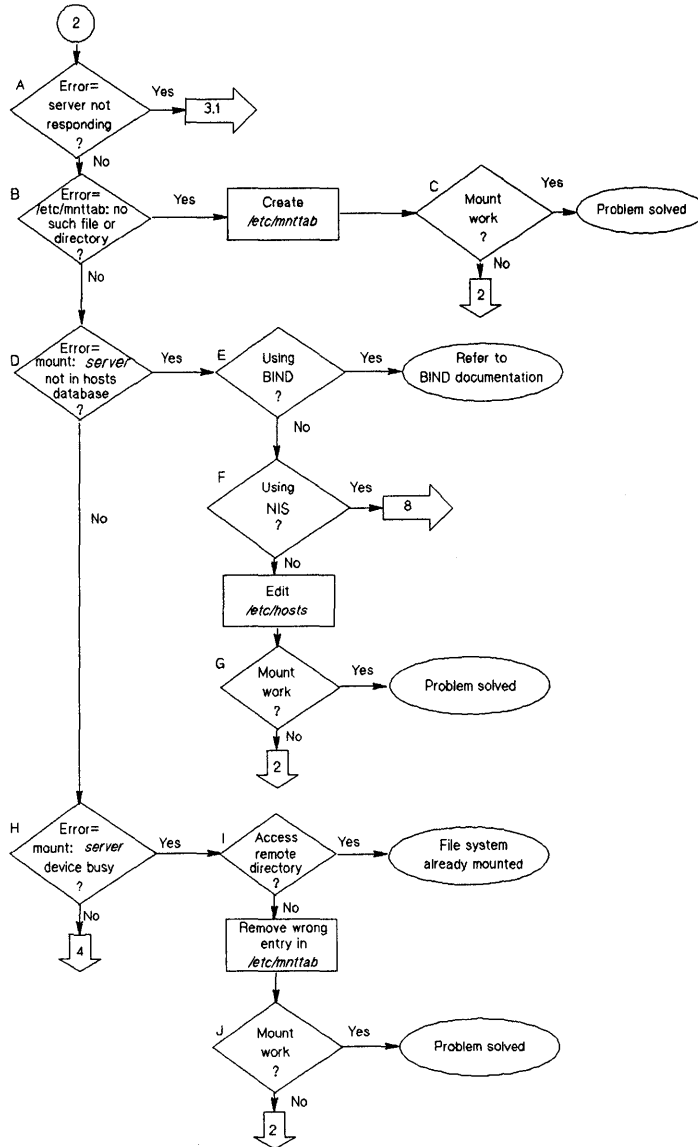
Flowchart 1: Initial Steps to Narrowing the Problem

Initial Steps to Narrowing the Problem (Flowchart 1)

Begin your troubleshooting efforts with Flowchart 1 since it helps you determine the best troubleshooting path based on the problem's symptoms.

Question	Yes: Action	No: Action
<p>A. Does the system hang during boot when mounting remote files?</p> <p>Systems hanging during boot where remote mounts generally occur may indicate one or more servers are down or the network connection to one or more servers is faulty.</p>	See Flowchart 2.	See A1.
A1. Are you using VHE?	See Flowchart 12.	See B.
B. Are you able to login?	See C.	You will receive error messages or the system will fail to respond if you cannot log into it. See D.
B1. Is the problem experienced while using the "on" command?	See Flowchart 18.	See C.
C. When trying to mount a remote directory, do error messages indicate the attempt failed?	See Flowchart 2.	See E.
D. Are you using NIS?	See Flowchart 7.	See I.
E. Do programs performing remote file accesses hang?	See Flowchart 5.	See F.
F. Does the system report unknown host errors during execution of commands or programs?	See G.	See H.
G. Are you using NIS?	See Flowchart 7.	Edit /etc/hosts to add remote host, and then see K.

Question	Yes: Action	No: Action
H. Does everything work, but slowly?	See Flowchart 6.	See J.
I. Do your login scripts perform NFS remote file accesses?	See Flowchart 2.	Refer to the system login information in the <i>System Administration Tasks</i> manual.
M. Does the following message occur? ESTALE	The file was removed by another user. NFS allows file removal at any time.	See L.
K. Does the system hang during boot?	Restart Flowchart 1.	Problem solved.
L. Are you receiving unpredictable results when executing programs or commands?	See M.	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.
M. Are the server and client clocks synchronized?	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.	Reset the clocks using the <code>date</code> command, and then see N.
N. Do you receive unpredictable results to commands or programs?	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.	Problem solved.



Flowchart 2: Mount Fails

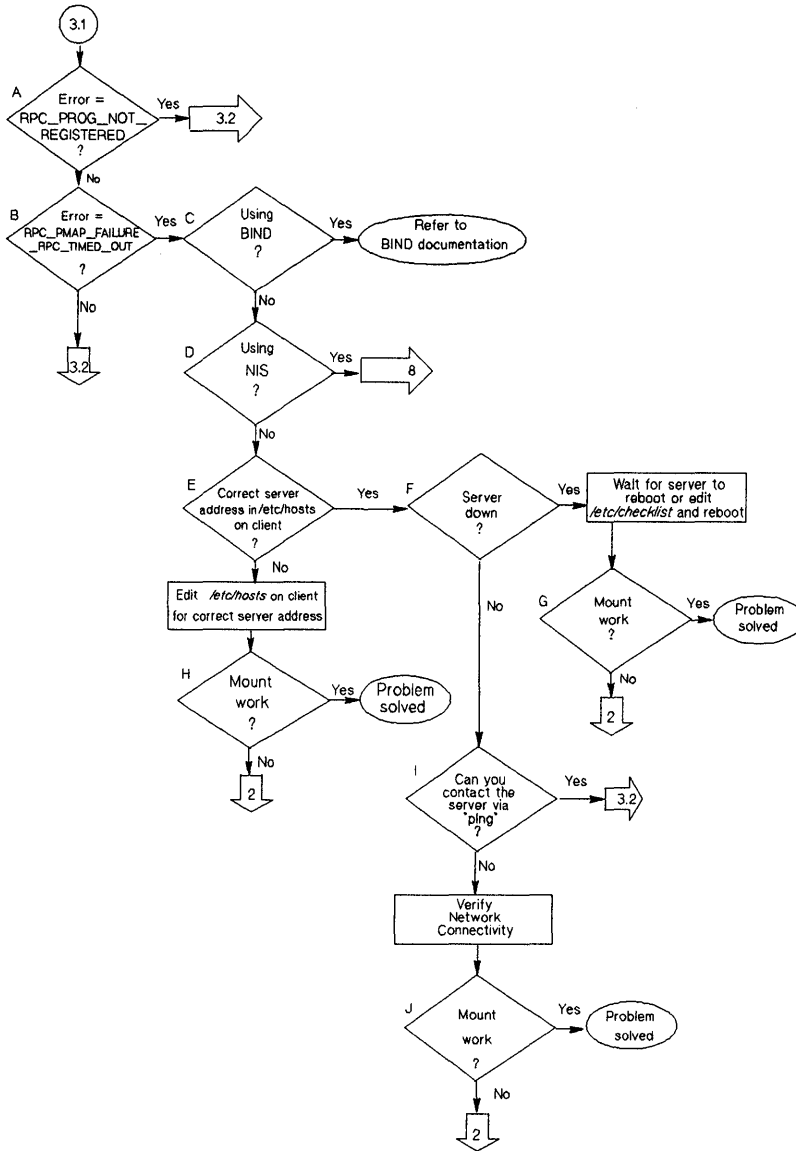
Mount Fails (Flowchart 2)

Use Flowchart 2 if your system hangs during the booting process when remote directories or files are mounted or if your remote mount attempts are unsuccessful.

Before using Flowchart 2, remember to check the mount command syntax and correct errors according to the error messages.

Question	Yes: Action	No: Action
A. Does the following error message occur on the client? server not responding	See Flowchart 3.1.	See B.
B. Does the following error message occur on the client? /etc/mnttab: no such file or directory	Create /etc/mnttab on the client by executing mount -u, and then see C. The system uses /etc/mnttab to log all mounted directories or files. <i>Note: Generally, at boot time /etc/rc creates /etc/mnttab.</i>	See D.
C. Can you mount the remote system?	Problem solved.	Restart Flowchart 2.
D. Does the following error message occur on the client? mount: server not in hosts database	See E.	See H.
E. Are you using BIND?	See the BIND documentation in <i>Installing and Administering ARPA Services</i> .	See F.
F. Are you using NIS?	See Flowchart 8.	Edit /etc/hosts on the client to include the desired remote host, and then see G.
G. Can you mount the remote system?	Problem solved.	Restart Flowchart 2.

Question	Yes: Action	No: Action
H. Does the following error message occur on the client? mount: server device busy	See I.	See Flowchart 4.
I. Can you access a remote directory in the desired remote directory?	You do not need to mount the directory since it is already mounted; problem solved.	On the client, unmount the directory you are trying to access for the remote directory you are trying to mount, and then see J.
J. Can you mount the remote system?	Problem solved.	Restart Flowchart 2.



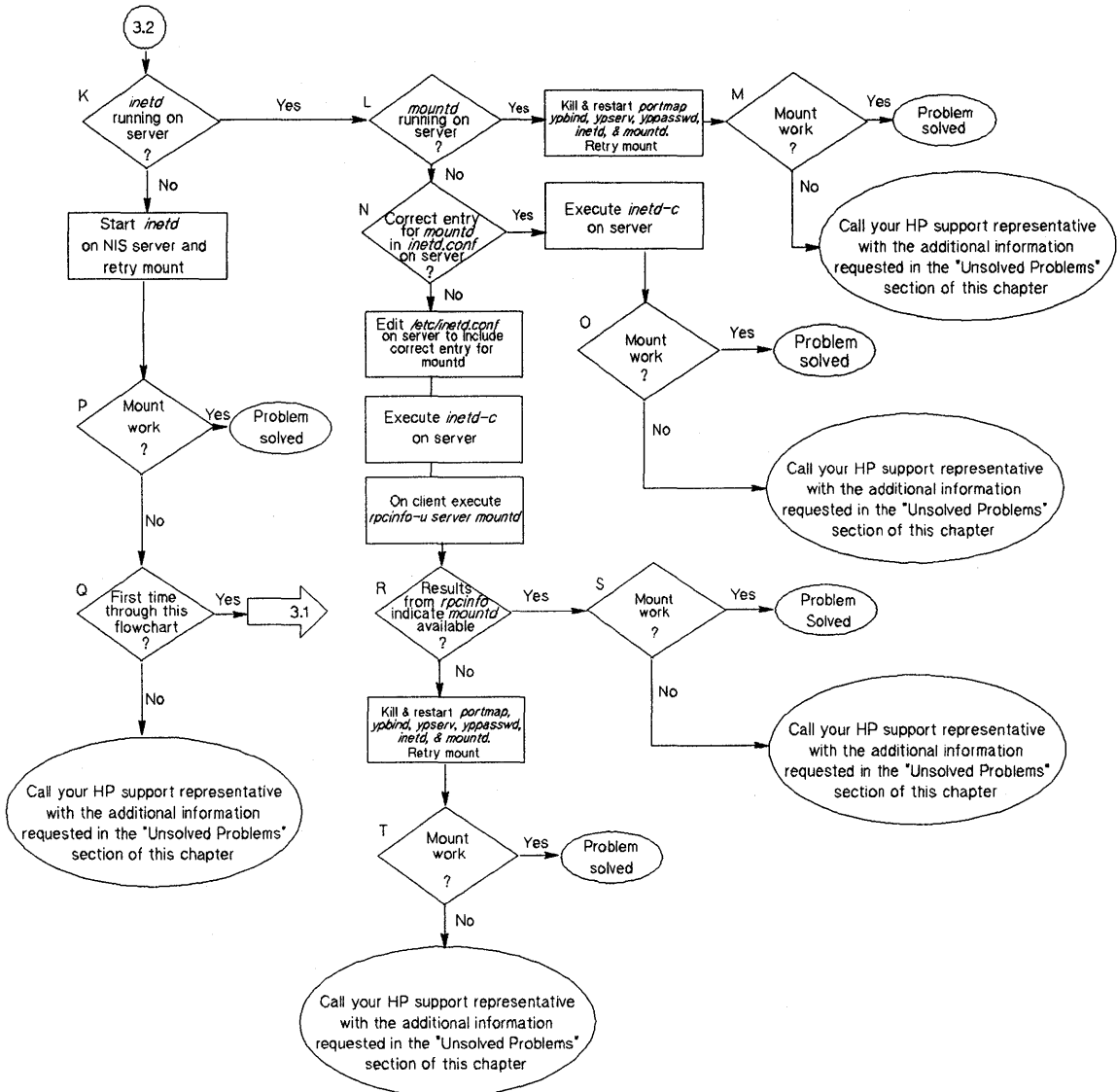
Flowchart 3.1: Server Not Responding

Server Not Responding (Flowchart 3.1)

This flowchart and corresponding instructions consist of two parts: Flowchart 3.1 and 3.2.

Question	Yes: Action	No: Action
A. Does the following error message occur? RPC_PROG_NOT_REGISTERED	See Flowchart 3.2.	See B.
B. Does the following error message occur? RPC_PMAP_FAILURE: RPC_TIMED_OUT	See C.	See Flowchart 3.2.
C. Are you using BIND?	See the BIND documentation in <i>Installing and Administering ARPA Services</i> .	See D.
D. Are you using NIS?	See Flowchart 8.	See E.
E. Is the server's address correct in the client's /etc/hosts?	See F.	Edit the client's /etc/hosts to include the correct address for the server you are trying to mount. See H.
F. Is the server you are trying to mount down? To check, ask your system administrator or try other network services to that system.	You have three options: - Do nothing on the system until the server reboots. - Unmount the files with the umount command. - Edit the client's /etc/checklist to remove the NFS entry for that server; reboot the system. See G.	See I.
G. Can you mount the remote system?	Problem solved.	See Flowchart 2.

Question	Yes: Action	No: Action
H. Can you mount the remote system?	Problem solved.	See Flowchart 2.
I. Can you contact the server using the ping diagnostic? Refer to the <i>Installing and Administering LAN</i> manuals for ping diagnostic information	See Flowchart 3.2.	Refer to the <i>Installing and Administering LAN</i> manual to verify link connectivity, and then see J.
J. Can you mount the remote system?	Problem solved.	See Flowchart 2.



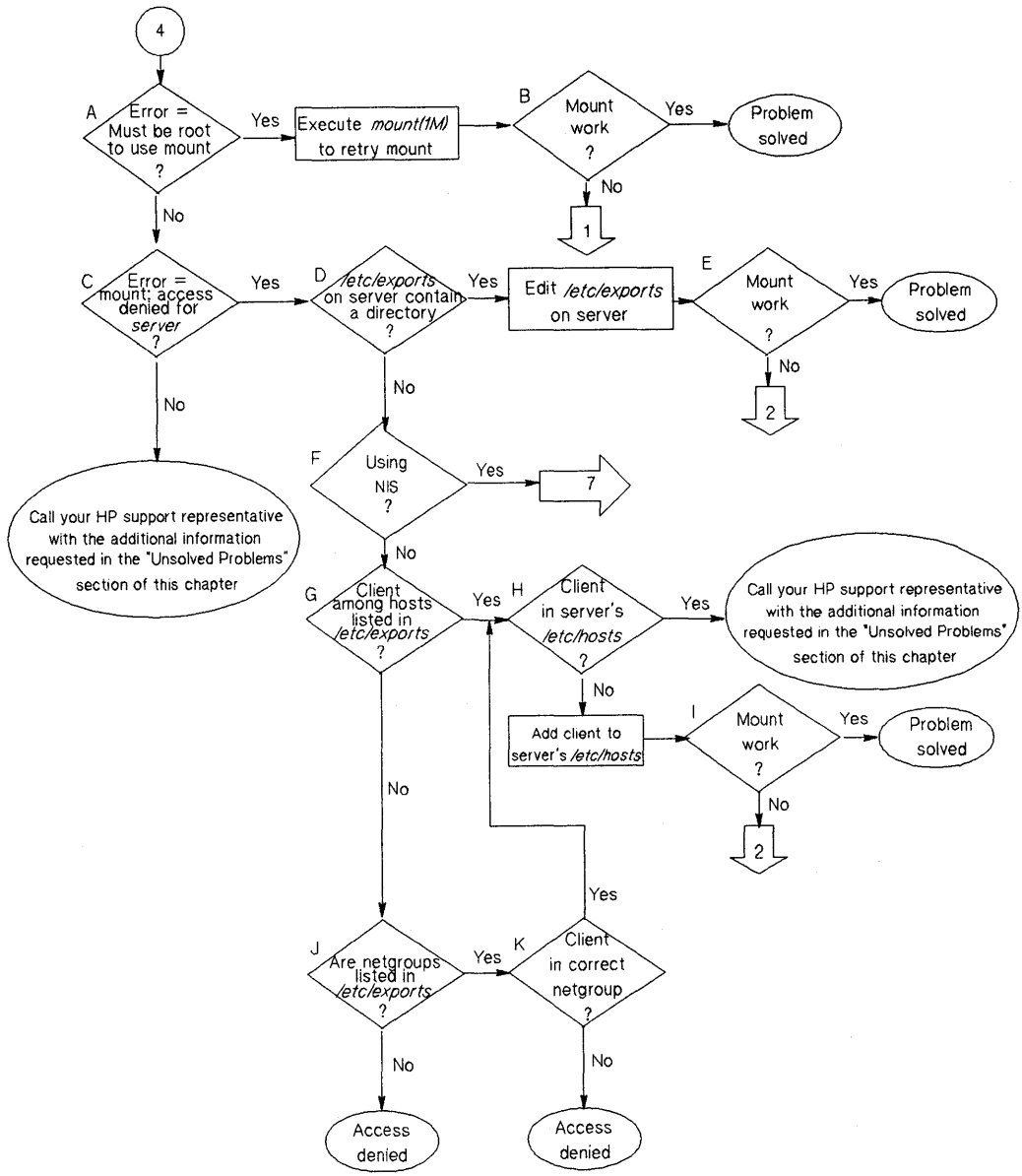
Flowchart 3.2: Server Not Responding

Server Not Responding (Flowchart 3.2)

Question	Yes: Action	No: Action
K. Is inetd running on the server?	See L.	Start <code>/etc/inetd</code> on the server, retry the mount, and then see P.
L. Is mountd running on the server?	Kill and restart the following daemons on the server in the order specified: <ul style="list-style-type: none"> - portmap - ypbind * - ypserv * - yppasswdd * - inetd -c if not configured in <code>/etc/inetd.conf</code> - mountd * only if using NIS Retry the mount, and then see M.	See N.
M. Can you mount the remote system?	Problem solved.	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.
N. Is the correct mountd entry in <code>inetd.conf</code> on the server? Ensure the entry is not commented out with a # (pound sign). Or is <code>START_mountd</code> set in <code>/etc/netnfsrc</code>	Make sure a mountd is running, if not, execute <code>/usr/etc/rpc.mountd</code> . Execute <code>inetd -c</code> on the server, and then see O.	<ol style="list-style-type: none"> 1. Edit the server's <code>/etc/inetd.conf</code> file to include the correct mountd entry. 2. Execute <code>inetd -c</code> on the server to read changes in <code>/etc/inetd.conf</code>. 3. Execute <code>rpcinfo -u</code> on the client. 4. See R.

Question	Yes: Action	No: Action
O. Can you mount the remote system?	Problem solved.	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.
P. Can you mount the remote system?	Problem solved.	See Q.
Q. Is this the first time you used this flowchart for this problem?	Restart Flowchart 3.1.	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.
R. Do the results from <code>rpcinfo -u</code> indicate a <code>mountd</code> process is available on the server?	See S.	<p>Kill and restart the following daemons on the server in the order specified:</p> <ul style="list-style-type: none"> - portmap - ybind * - ypserv * - yppasswdd * - inetd -c if not configured in <code>/etc/inetd.conf</code> - mountd <p>* only if using NIS</p> <p>Retry the mount, and then see T.</p>
S. Can you mount the remote system?	Problem solved.	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.

Question	Yes: Action	No: Action
T. Can you mount the remote system?	Problem solved.	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.

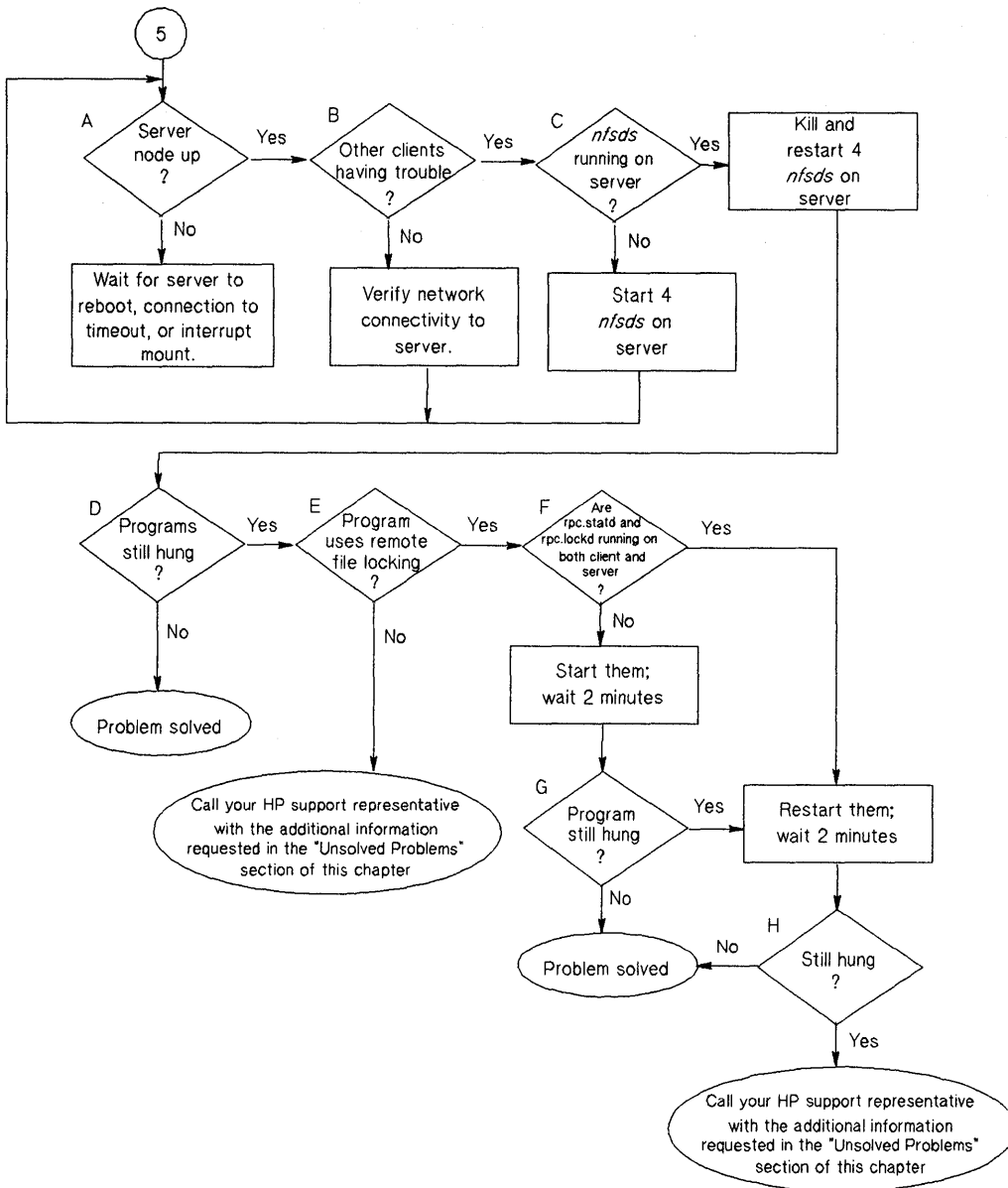


Flowchart 4: Restricted Access

Restricted Access (Flowchart 4)

Question	Yes: Action	No: Action
<p>A. Does the following error message occur on the client?</p> <p>Must be root to use mount</p>	<p>Log in as superuser, execute mount, and then see B.</p>	<p>See C.</p>
<p>B. Can you mount the remote system?</p>	<p>Problem solved.</p>	<p>See Flowchart 1.</p>
<p>C. Does the following error message occur on the client?</p> <p>mount: access denied for server</p>	<p>See D.</p>	<p>Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.</p>
<p>D. Is the directory exported?</p>	<p>Make sure the directory is exported by executing showmount -c or exportfs on the server and then see E.</p>	<p>See F.</p>
<p>E. Can you mount the remote system?</p>	<p>Problem solved.</p>	<p>See Flowchart 2.</p>
<p>F. Are you using NIS?</p>	<p>See Flowchart 7.</p>	<p>See G.</p>
<p>G. If hosts are listed in /etc/exports, is the client among the hosts listed for the desired file system?</p>	<p>See H.</p>	<p>See J.</p>
<p>H. Is the client listed in the server's /etc/hosts?</p>	<p>Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.</p>	<p>Add client to server's /etc/hosts</p>
<p>I. Can you mount the remote system?</p>	<p>Problem solved.</p>	<p>See Flowchart 2.</p>

Question	Yes: Action	No: Action
J. Are netgroups listed for this file system in server's exportentlist?	See K.	Access for this client is deliberately denied.
K. Is the client listed in the appropriate netgroup for this file system in /etc/netgroup?	See H.	Access for this client is deliberately denied.



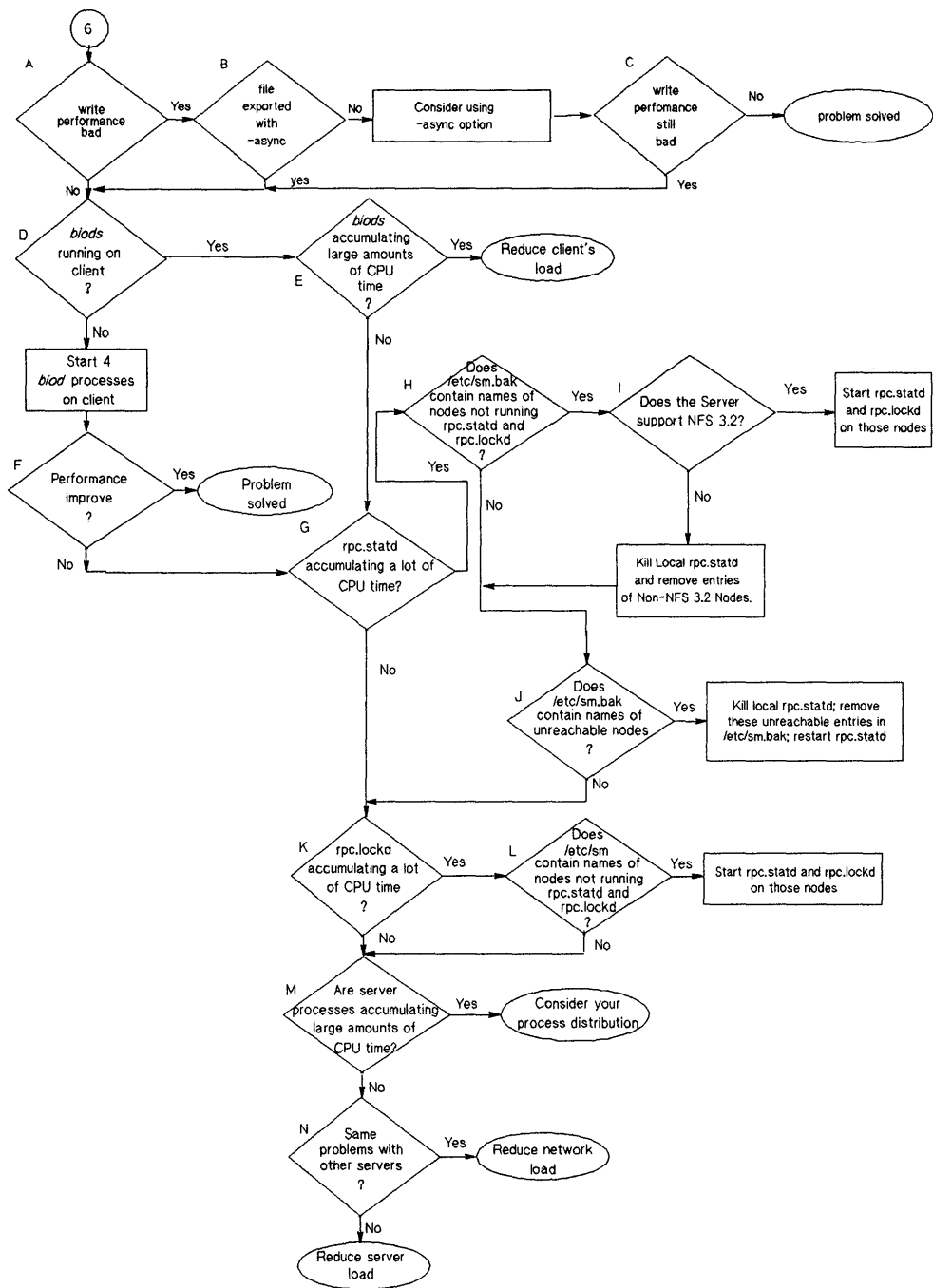
Flowchart 5: Programs Hang

Programs Hang (Flowchart 5)

Programs are most likely to hang if network communication is lost to the server, if the server is down, or if daemons are hung.

Question	Yes: Action	No: Action
A. Is the server node running?	See B.	For hard mounts, do <i>one</i> of the following: <ul style="list-style-type: none">- Wait for the server to reboot.- Interrupt the mount. For soft mounts, wait for the mount to time out. See A.
B. Are other client nodes having trouble?	See C.	Verify the network connectivity. Refer to the <i>Installing and Administering LAN</i> manuals. See A.
C. Are nfsd daemons running on the server?	Kill and restart four nfsd daemons on the server, and then see D.	Start four nfsd daemons on the server, and then see A.
D. Do the programs hang?	See E.	Problem solved.
E. Does the program use remote file locking?	See F.	Call your HP Support representative with the additional information requested in the Unsolved Problems section of this chapter.

Question	Yes: Action	No: Action
<p>F. Are rpc.statd and rpc.lockd running on both the client and the server?</p>	<p>Start rpc.statd first, then start rpc.lockd.</p> <p>Restart them and wait 2 minutes.</p> <p>See H.</p>	<p>Start them and wait 2 minutes.</p> <p>See G.</p>
<p>G. Does the program still hang?</p>	<p>Restart rpc.statd and rpc.lockd on both the client and the server. Wait 2 minutes.</p> <p>See H.</p>	<p>Problem solved.</p>
<p>H. Does the program still hang?</p>	<p>Call your HP Support representative with the additional information requested in the Unsolved Problems section of this chapter.</p>	<p>Problem solved.</p>



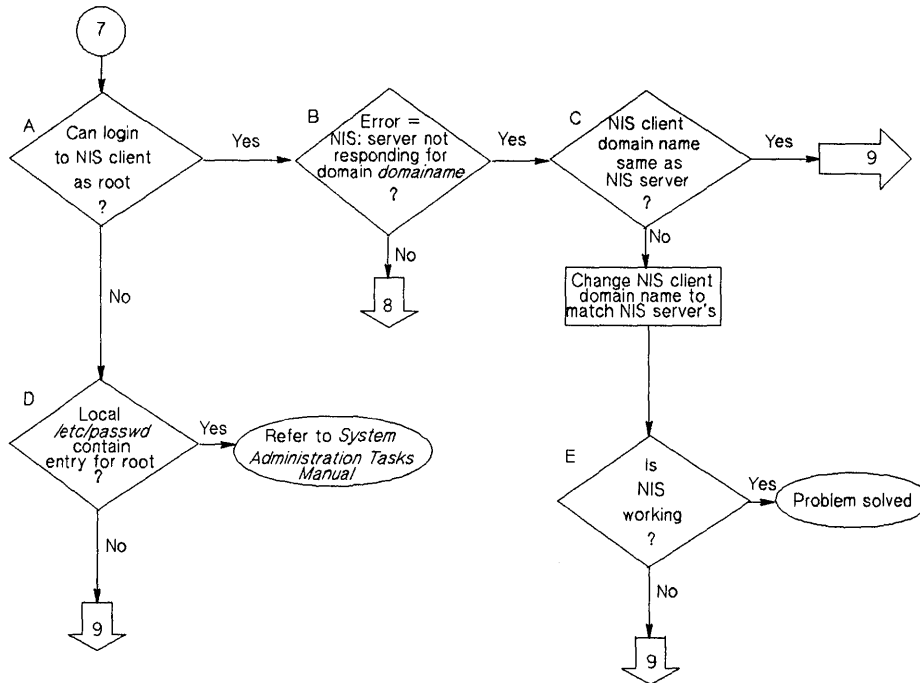
Flowchart 6: Performance Problems

Performance Problems (Flowchart 6)

Question	Yes: Action	No: Action
A. Is the write performance bad?	See B.	See D.
B. Is the file exported with -async?	See D	Consider using -async option, then see C.
C. Is the write performance still bad?	See D.	Problem Solved.
D. Are the biods running on the client?	See E.	Start four biod processes on the client, and then see F.
<p>E. Are the client biod daemons accumulating large amounts of CPU time?</p> <ol style="list-style-type: none"> 1. List the client processes using ps. 2. Copy a large file to the server system, and list the client biod processes again. 3. Compare the CPU time for the biod processes before and after the file copy. 	Reduce the client's load to fewer NFS transactions by reducing the number of users or storing more files locally.	See G.
F. Has the performance improved?	Problem solved.	See G.
G. Is rpc.statd accumulating a lot of CPU time? (On the client?)	See H.	See K.
H. Does /etc/sm.bak contain names of nodes not running rpc.statd and rpc.lockd?	See I.	See J.

Question	Yes: Action	No: Action
I. Does the server support NFS 3.2 functionality? (HP-UX 6.5 or later for the Series 300/400. HP-UX 7.0 or later for other HP architectures.)	Start rpc.statd and rpc.lockd on those nodes.	Kill local rpc.statd and rpc.lockd and remove entries of non-NFS 3.2 nodes, then see J.
J. Does /etc/sm.bak contain names of unreachable nodes?	Kill local rpc.statd and rpc.lockd, remove these unreachable entries in /etc/sm.bak, and restart rpc.statd and rpc.lockd	See K.
K. Is rpc.lockd accumulating a lot of CPU time? (On the client?)	See L.	See M.
L. Does /etc/sm.bak contain names of nodes not running rpc.statd and rpc.lockd?	Start rpc.statd and rpc.lockd on those nodes.	See M.
M. Are processes on the server accumulating large amounts of CPU time (especially nfsd, inetd, and portmap)?	Consider whether you need to distribute your processing by adding additional systems.	See N.
N. Are the same performance problems evident with other servers?	Reduce the network load.	Reduce the server's load by adding more servers.

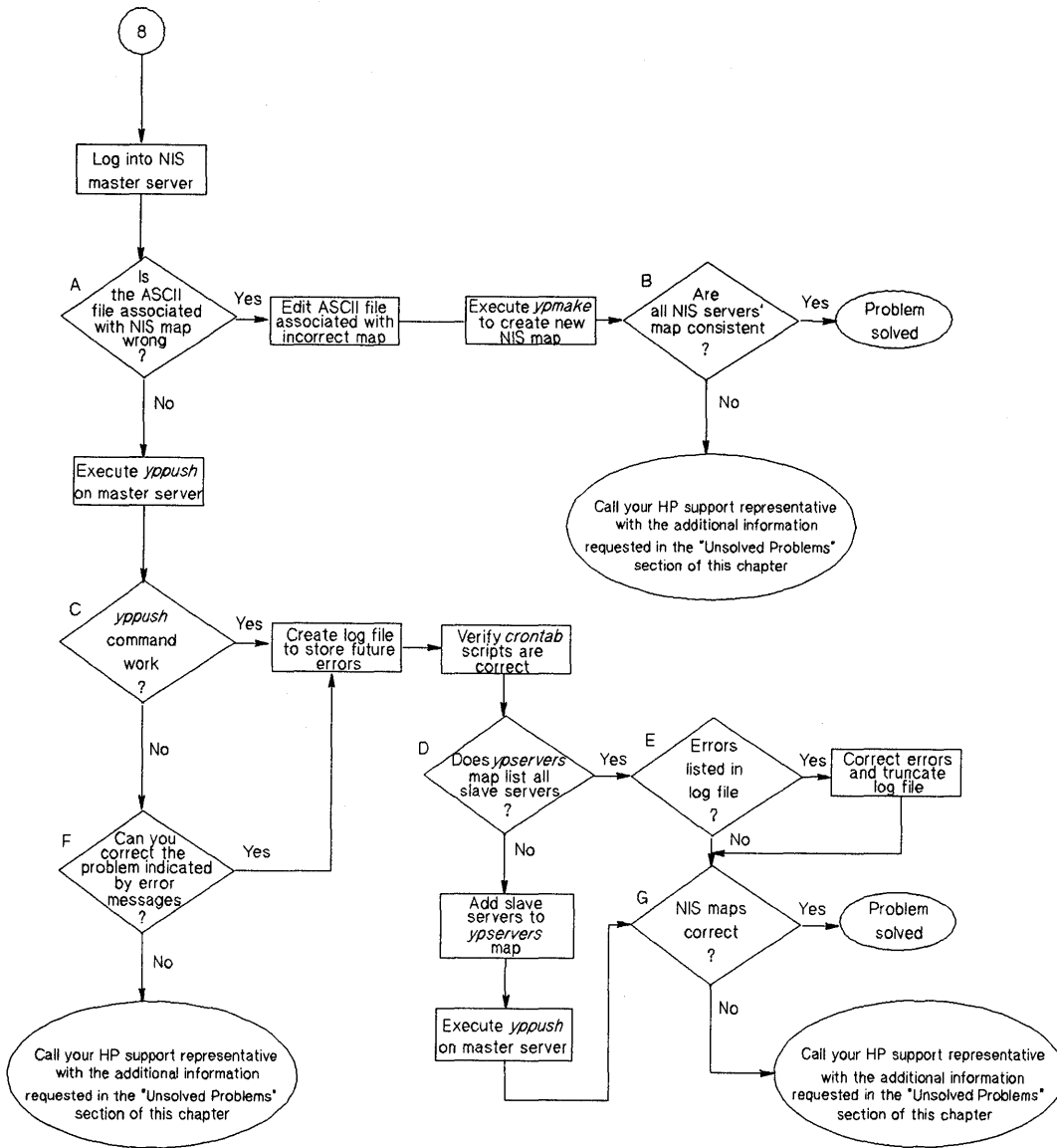
Troubleshooting NIS



Flowchart 7: Initial Steps to Troubleshooting NIS

Initial Steps to Troubleshooting NIS (Flowchart 7)

Question	Yes: Action	No: Action
A. Can you login as root on the NIS client?	See B.	See D.
B. Does the following error message occur on the console or in the ypbind log file? NIS: server not responding for domain domain_name	See C.	See Flowchart 8.
C. Is the NIS client's NIS domain name the same as the NIS server's?	See Flowchart 9.	Change the NIS client's NIS domain name to be the same as the NIS server's, and then see E. domainname domain_name
D. Does the local /etc/passwd file contain an entry for root?	The problem is not associated with NIS or NFS. Refer to the <i>System Administration Tasks Manual</i> .	You cannot log into the NIS client until NIS is functioning unless you have an entry for a user in the local /etc/passwd file. See Flowchart 9.
E. Is NIS working? If you can access the NIS server's maps using ypcat or ypmatch	NIS is probably functioning correctly.	Problem solved.



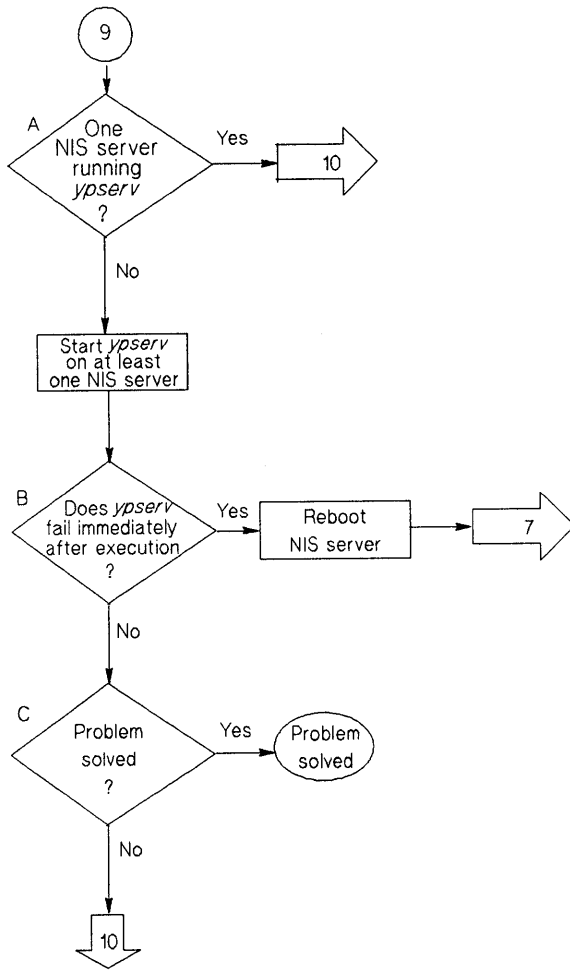
Flowchart 8: Incorrect NIS Maps

Incorrect NIS Maps (Flowchart 8)

Log into the NIS master server as root before starting Flowchart 8.

Question	Yes: Action	No: Action
<p>A. On the NIS master server, does the ASCII file associated with the NIS map need to be updated (e.g., update /etc/hosts)?</p>	<ol style="list-style-type: none"> 1. Edit the ASCII file associated with the incorrect NIS map. 2. Execute ypmake to create and distribute a new map to the NIS slave servers. 3. See B. 	<p>Execute yppush on the NIS master server, and then see C.</p> <p>yppush map_name</p>
<p>B. Are all NIS server's maps consistent? You can determine this by executing yppoll and then comparing order numbers.</p>	<p>Problem solved.</p>	<p>Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.</p>
<p>C. Does yppush work correctly? If you do not receive error messages associated with the command, it probably executed successfully.</p>	<ol style="list-style-type: none"> 1. Create the log file /usr/etc/yp/ypxfr.log to trap future errors associated with yppush on each NIS slave server. 2. Verify that crontab scripts (on each slave server) copying the maps are correct. 3. See D. 	<p>See F.</p>
<p>D. Does the ypservers map list all NIS slave servers?</p> <p>ypcat -k ypservers</p>	<p>See E.</p>	<ol style="list-style-type: none"> 1. Add any missing NIS slave server to the ypservers map. 2. Execute yppush on the NIS master server to update all NIS slave servers. 3. See G.

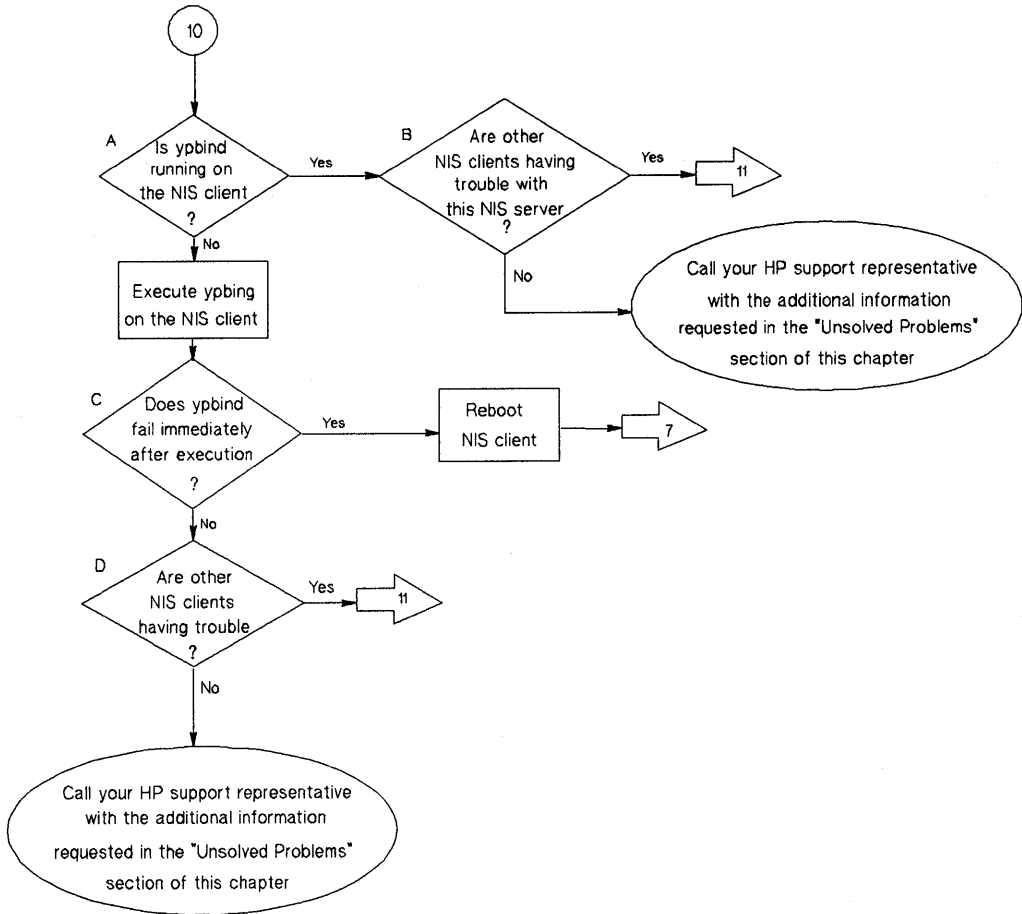
Question	Yes: Action	No: Action
E. Does <code>/usr/etc/yp/ypxfr.log</code> on the slave server list errors?	Correct the errors, truncate the log file, and then see G.	See G.
F. Can you correct the problem indicated by the error message?	<ol style="list-style-type: none"> 1. Create the log file <code>/usr/etc/yp/ypxfr.log</code> to trap future errors associated with <code>yppush</code> on each NIS slave server. 2. Verify that <code>crontab</code> scripts (on each slave server) distributing the maps are correct. 3. See D. 	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.
G. Are the NIS maps correct?	Problem solved.	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.



Flowchart 9: ypserv Problems

ypserv Problems (Flowchart 9)

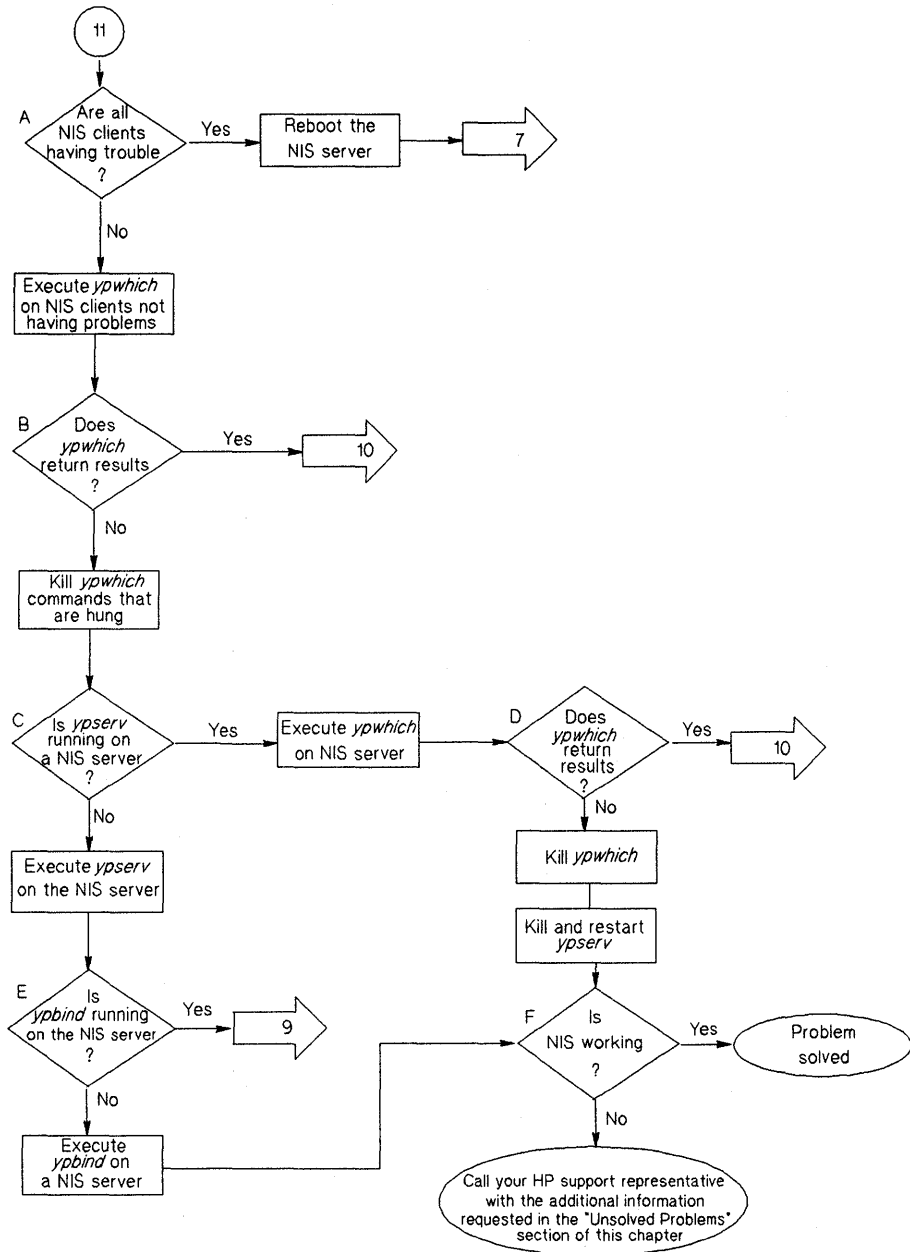
Question	Yes: Action	No: Action
A. Is at least one NIS server in the NIS domain running ypserv?	See Flowchart 10.	Start ypserv on at least one NIS server in the NIS domain, and then see B.
B. Does ypserv fail immediately after starting it?	Reboot the NIS server, and then see Flowchart 7.	See C.
C. Is the problem solved?	Problem solved.	See Flowchart 10.



Flowchart 10: ypbind Problems

ypbind Problems (Flowchart 10)

Question	Yes: Action	No: Action
A. Is ypbind running on the NIS client?	See B.	Execute ypbind on the NIS client, and then see C.
B. Are other NIS clients having trouble with this NIS server?	See Flowchart 11.	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.
C. Does ypbind crash immediately after starting it?	Reboot the NIS client, and then see Flowchart 7.	See D.
D. Are other NIS clients having trouble with this NIS server?	See Flowchart 11.	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.

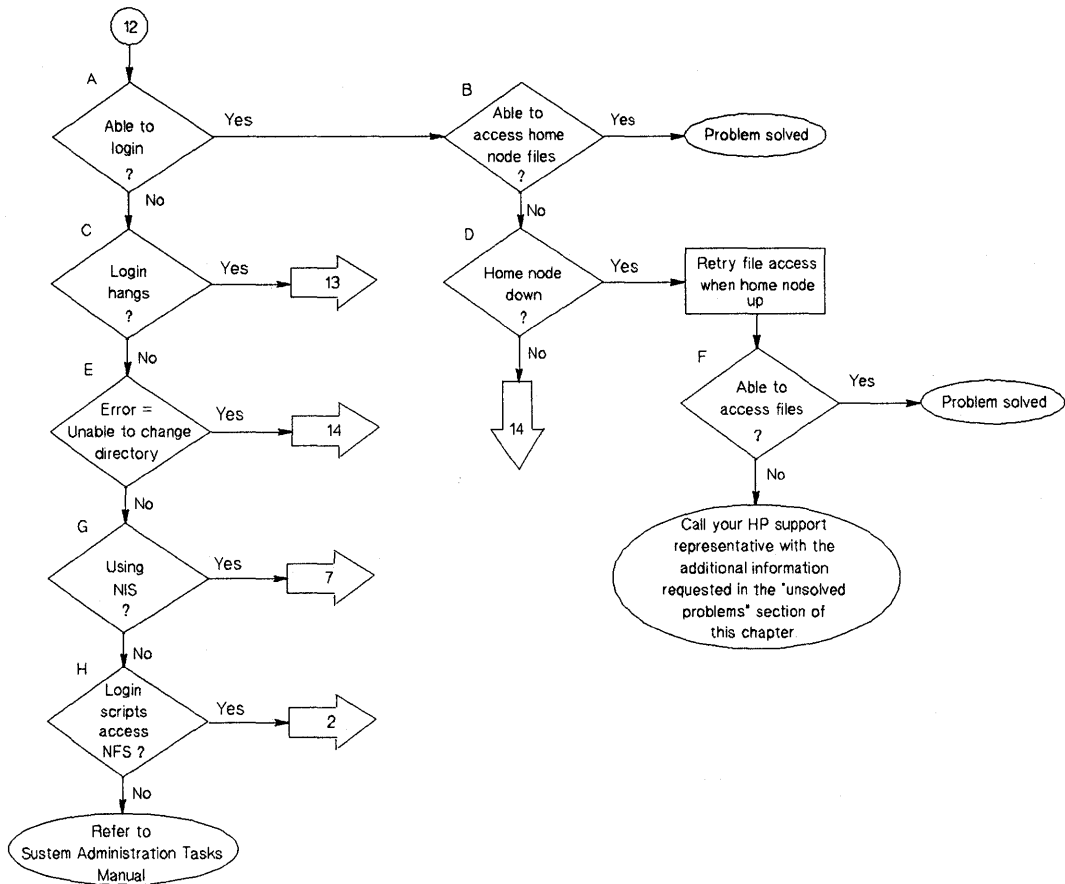


Flowchart 11: Multiple NIS Client Problems

Multiple NIS Client Problems (Flowchart 11)

Question	Yes: Action	No: Action
A. Are all NIS clients having trouble with this NIS server?	Reboot the NIS server, and then see Flowchart 7.	Execute ypwhich on the NIS client nodes not having problems, and then see B.
B. Does the ypwhich command return results on the NIS client?	See Flowchart 10.	Kill ypwhich commands that are hung on NIS clients, and then see C.
C. Is ypserv running on the NIS server?	Execute ypwhich on the NIS server, and then see D.	Execute ypserv on the NIS server, and then see E.
D. Does ypwhich return results on the NIS server?	See Flowchart 10.	<ol style="list-style-type: none"> 1. Kill ypwhich on the NIS server. 2. Kill and restart ypserv. 3. See F.
E. Is ypbind running on the NIS server?	See Flowchart 9.	Execute ypbind on the NIS server, and then see F.
F. Is NIS functioning correctly on all NIS clients?	Problem solved.	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.

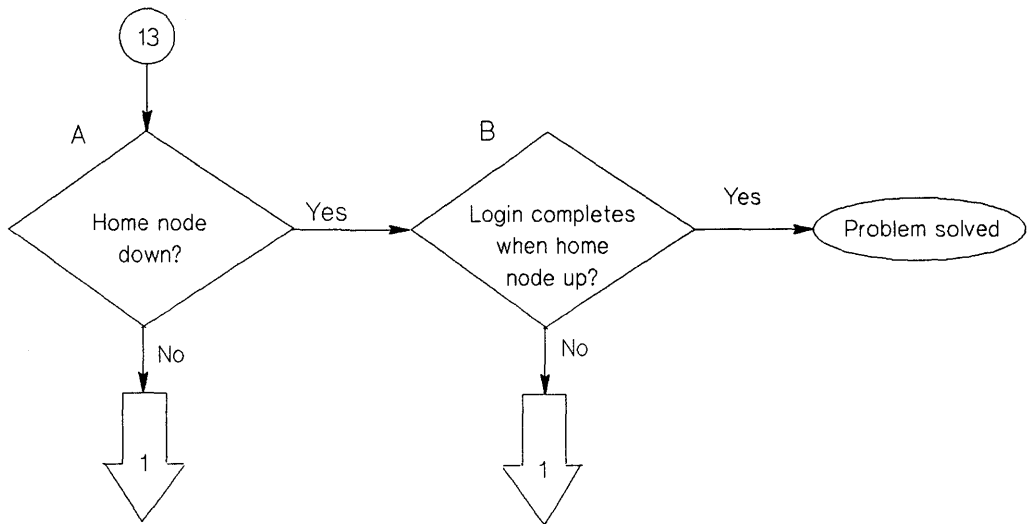
Troubleshooting VHE



Flowchart 12: Initial Steps to Troubleshooting VHE

Initial Steps to Troubleshooting VHE (Flowchart 12)

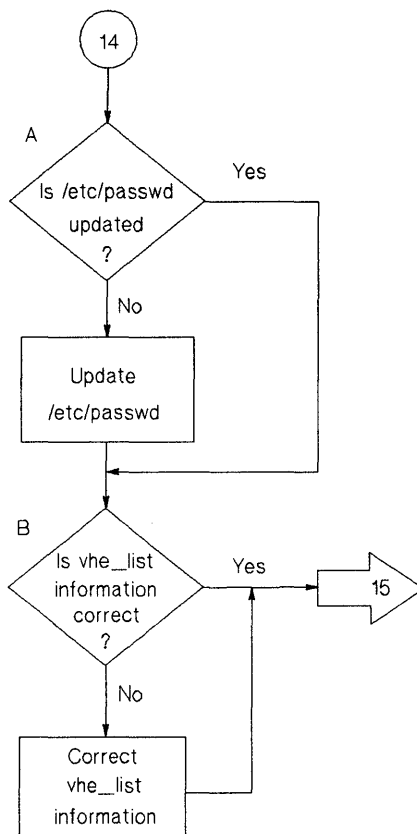
Question	Yes: Action	No: Action
A. Are you able to log in?	See B.	See C.
B. Are you able to access files on the home node?	No problem.	See D.
C. Does the machine hang during login?	See Flowchart 13.	See E.
D. Is the home node down?	Retry accessing files when the home node is up; then see F.	See Flowchart 14.
E. Do you receive the following error message? Unable to change directory to home directory	See Flowchart 14.	See G.
F. Are you able to access files on the home node?	Problem solved.	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.
G. Are you using NIS?	See Flowchart 7.	See H.
H. Do your login scripts perform NFS remote file access?	See Flowchart 2.	The problem is probably unassociated with the network services. Refer to the system login information in the <i>System Administration Tasks</i> manual.



Flowchart 13: Home Node Goes Down After Mount Complete

Home Node Goes Down After Mount Complete (Flowchart 13)

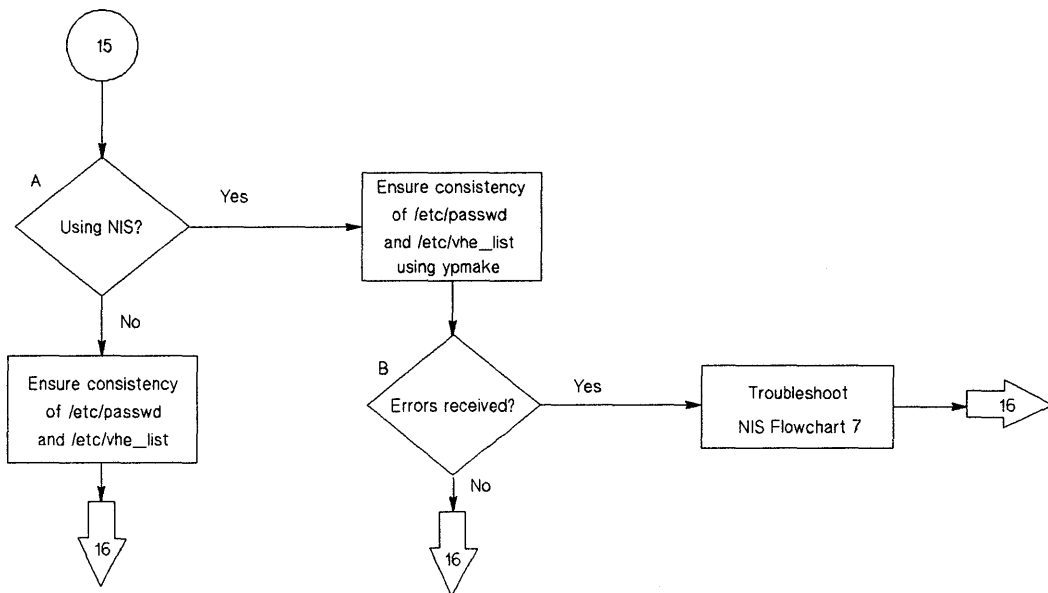
Question	Yes: Action	No: Action
A. Is the home node down?	Try logging in again once the home node comes up; then see B.	See Flowchart 1.
B. Does the login complete once the home node comes up?	Problem solved.	See Flowchart 1.



Flowchart 14: Checking /etc/passwd and /etc/vhe_list Files

Checking /etc/passwd and /etc/vhe_list Files (Flowchart 14)

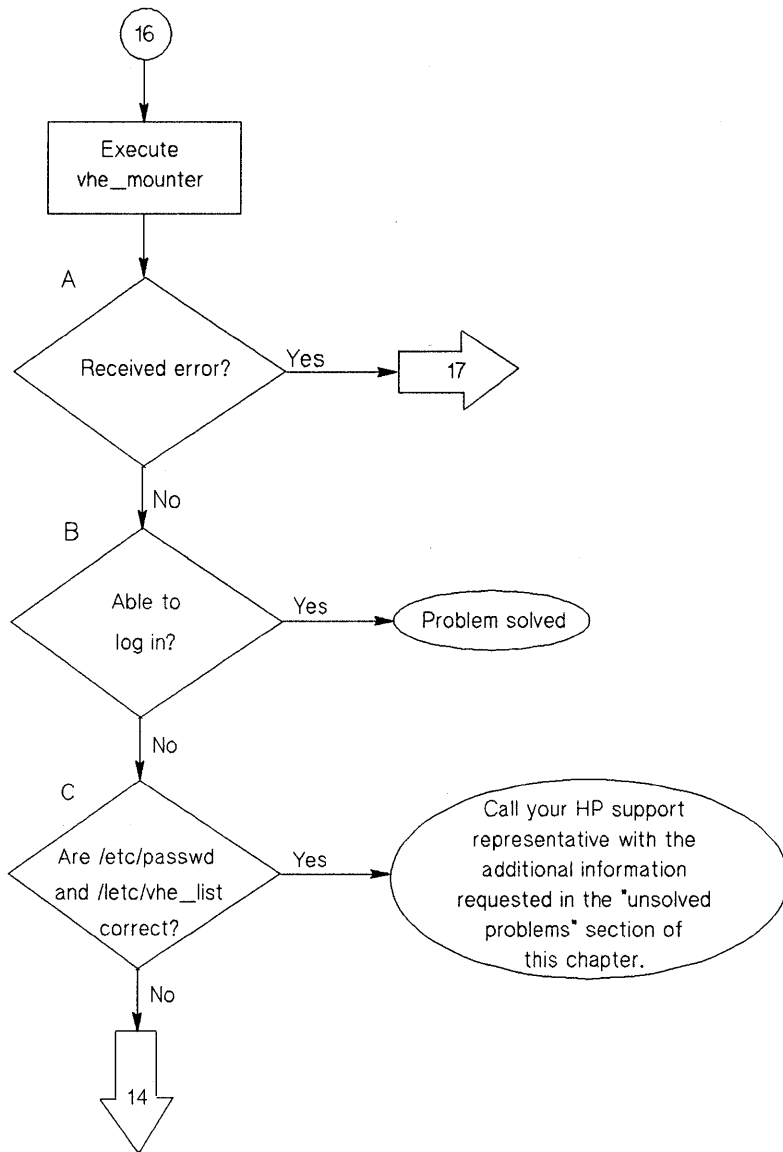
Question	Yes: Action	No: Action
A. Is the /etc/passwd file updated to prefix the home directory with the NFS mount point?	See B.	Update the /etc/passwd file as described in the VHE Configuration and Maintenance chapter; go to B.
B. Is the information in the /etc/vhe_list file correct?	See Flowchart 15.	Correct the /etc/vhe_list file information; see Flowchart 15.



Flowchart 15: Consistency of /etc/passwd and /etc/vhe_list

Consistency of /etc/passwd and /etc/vhe_list (Flowchart 15)

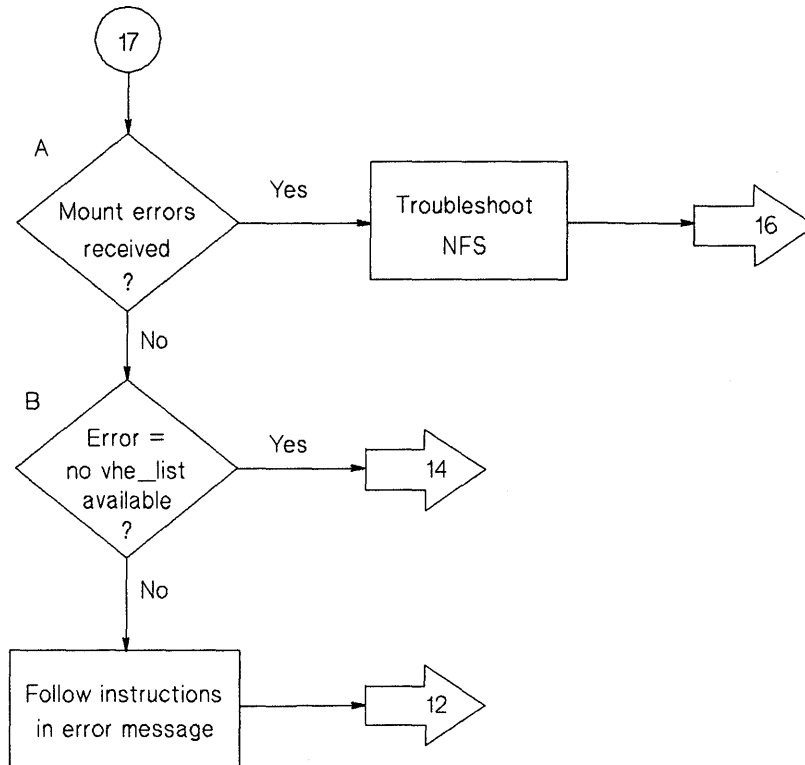
Question	Yes: Action	No: Action
A. Are you using the Network Information Service (NIS) to ensure consistency of /etc/passwd and /etc/vhe_list information?	Ensure consistency of the /etc/passwd and /etc/vhe_list files on all nodes in the VHE group by executing the following command: /usr/etc/yp/ypmake passwd vhe_list See B.	Ensure consistency of the /etc/passwd and /etc/vhe_list files on all nodes in the VHE group. See Flowchart 16.
B. Did you receive any errors when executing ypmake?	Go to the NIS Flowchart 7 and complete troubleshooting steps; then return to VHE Flowchart 16.	Go to Flowchart 16.



Flowchart 16: Execution of vhe-mounter

Execution of vhe_mounter (Flowchart 16)

Question	Yes: Action	No: Action
A. Did you receive any errors while executing vhe_mounter?	See Flowchart 17.	See B.
B. Are you able to log in?	Problem Solved.	See C.
C. Is the information for the home node entered into the /etc/passwd and /etc/vhe_list files?	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.	

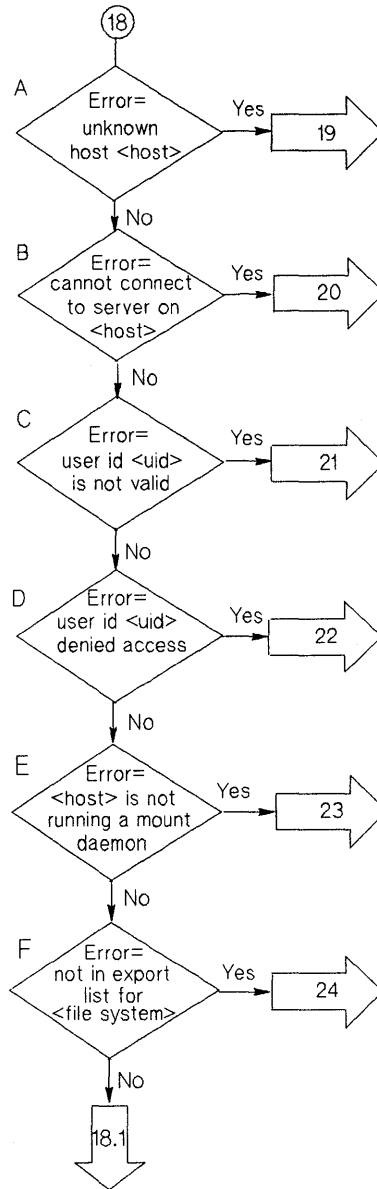


Flowchart 17: Error Message from vhe_mounter

Error Message from vhe_mounter (Flowchart 17)

Question	Yes: Action	No: Action
A. Were any mount errors encountered (mount errors begin with mount:)?	Troubleshoot NFS (Flowchart 1); then see Flowchart 16.	See B.
B. Does the following error message occur? no vhe_list available	See Flowchart 14.	If an error message other than those mentioned is printed, follow the instructions in that error message; then re-enter Flowchart 12 to see if problem is solved.

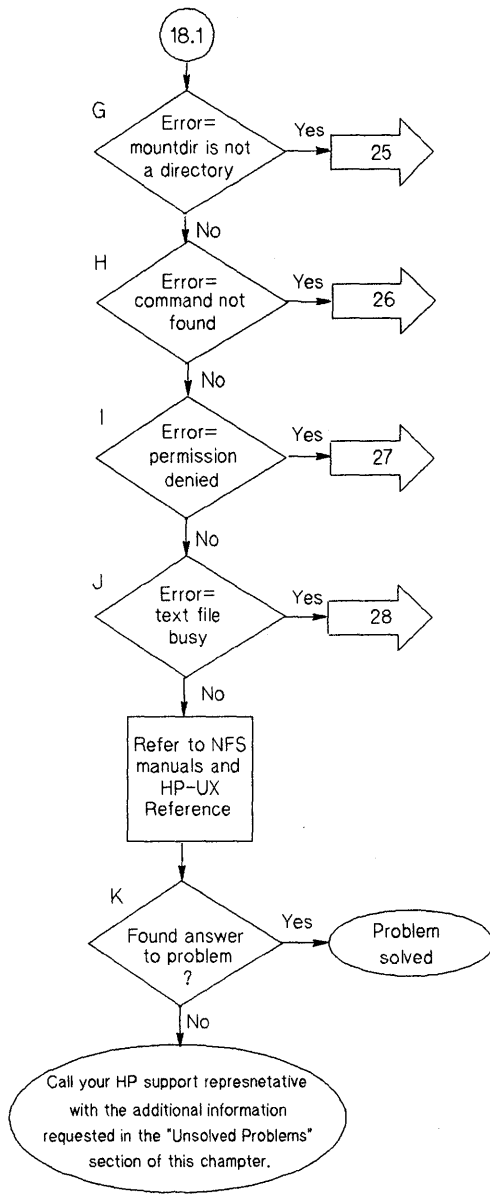
Troubleshooting REX



Flowchart 18: Initial Steps to Troubleshoot REX

Initial Steps to Troubleshoot REX (Flowchart 18)

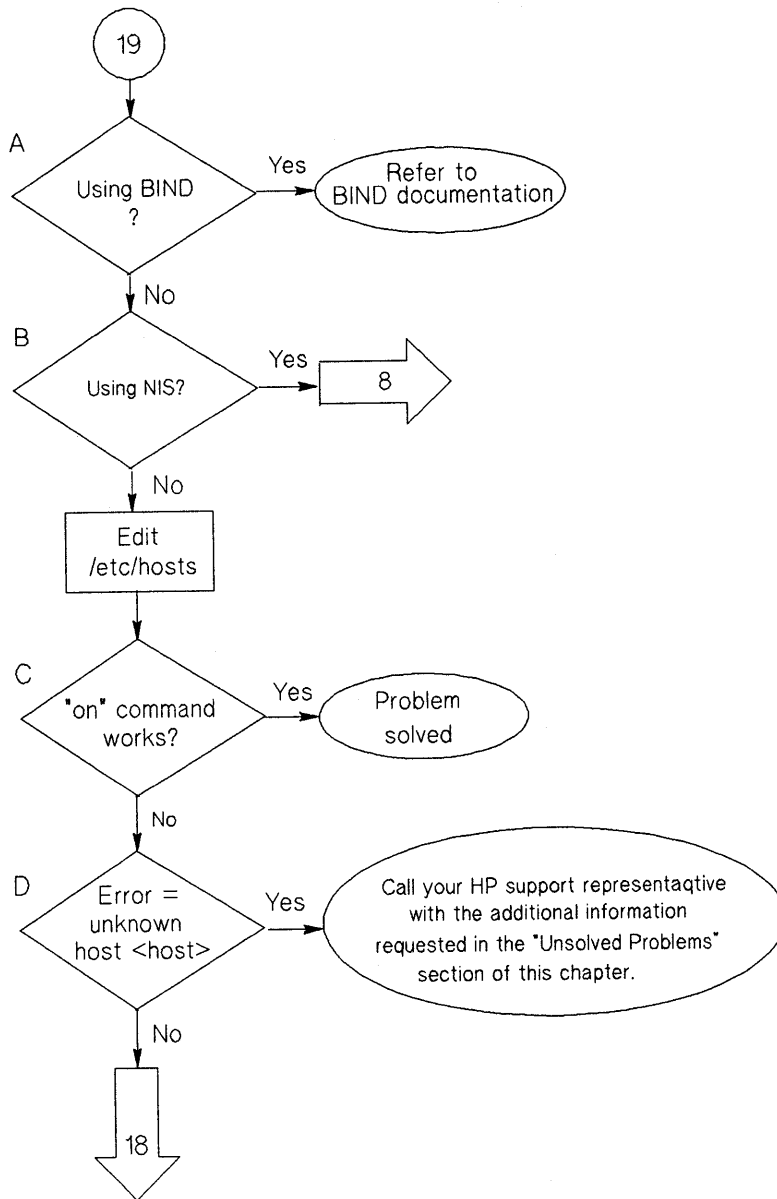
Question	Yes: Action	No: Action
<p>A. Does the following error message appear?</p> <p>on: unknown host < host ></p>	See Flowchart 19.	See B.
<p>B. Does the following error message appear?</p> <p>on: cannot connect to server on < host ></p>	See Flowchart 20.	See C.
<p>C. Does the following error message appear?</p> <p>on: rexd: user id is not valid</p>	See Flowchart 21.	See D.
<p>D. Does the following error message appear?</p> <p>on < server > : rexd: user id < uid > denied access</p>	See Flowchart 22.	See E.
<p>E. Does the following error message appear?</p> <p>on: < server > rexd: < host > is not running a mount daemon</p>	See Flowchart 23.	See F.
<p>F. Does the following error message appear?</p> <p>on < server > : rexd: not in export list for filesystem</p>	See Flowchart 24.	See Flowchart 18.1.



Flowchart 18.1: Initial Steps to Troubleshoot REX

Initial Steps to Troubleshoot REX (Flowchart 18.1)

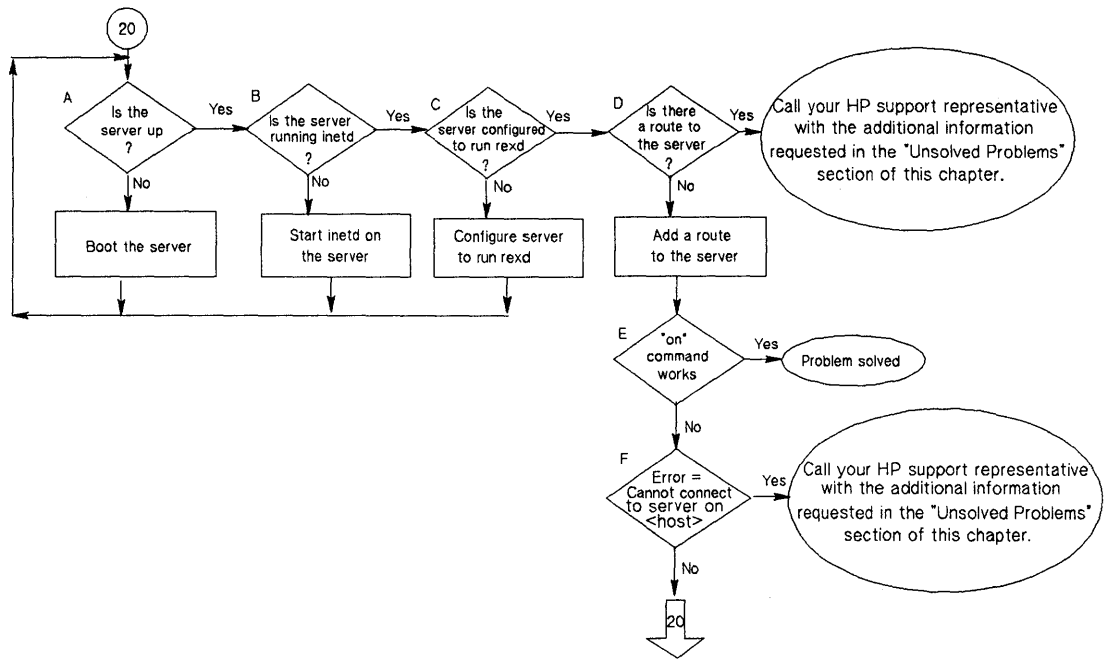
Question	Yes: Action	No: Action
<p>G. Does the following error message appear?</p> <p>on < server > : rexd: (< mountdir >) is not a directory</p>	See Flowchart 25.	See H.
<p>H. Does the following error message appear?</p> <p>on < server > : rexd: command not found</p>	See Flowchart 26.	See I.
<p>I. Does the following error message appear?</p> <p>on < server > : rexd: permission denied</p>	See Flowchart 27.	See J.
<p>J. Does the following error message appear?</p> <p>on < server > : rexd: text file busy</p>	See Flowchart 28.	<p>Refer to NFS manuals and <i>HP-UX Reference</i>.</p> <p>See K.</p>
<p>K. Found answer to your problem?</p>	Problem solved.	<p>Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.</p>



Flowchart 19: Unknown Host

Unknown Host (Flowchart 19)

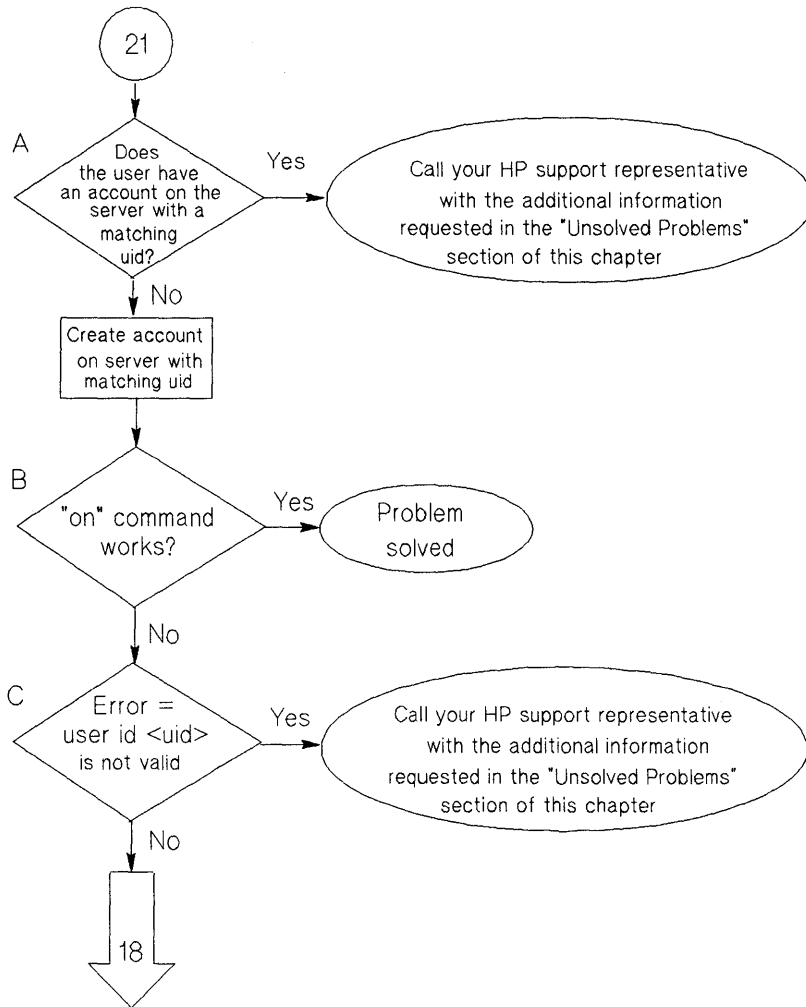
Question	Yes: Action	No: Action
A. Is your node using BIND?	Refer to BIND documentation in <i>Installing and Administering ARPA Services</i> .	See B.
B. Is your node using the Network Information Service (NIS)?	See Flowchart 8.	Edit /etc/hosts on the client to include the desired remote host. See C.
C. <code>on</code> command works now?	Problem solved.	See D.
D. Does the following error message appear? <code>on: unknown <host ></code>	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.	See Flowchart 18.



Flowchart 20: Cannot Connect to REX Server

Cannot Connect to REX Server (Flowchart 20)

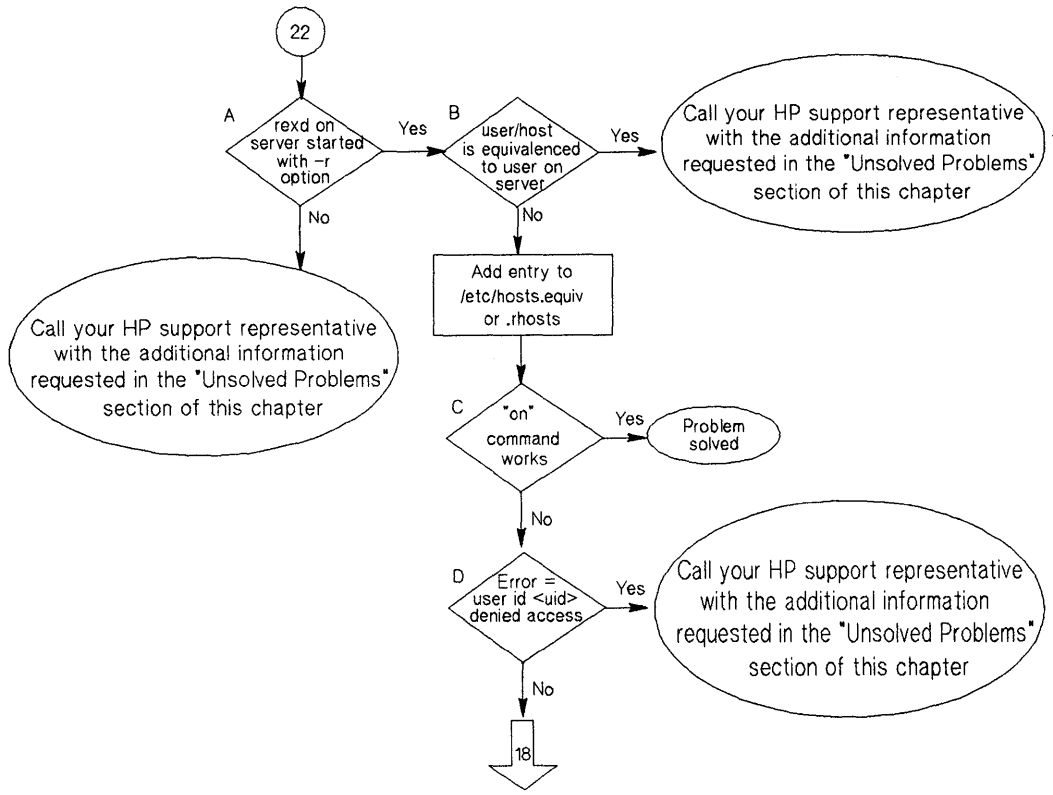
Question	Yes: Action	No: Action
A. Is the rex server node up?	See B.	Boot the rex server node. See A.
B. Is the rex server node running inetd?	See C.	Start inetd on the rex server node. See A.
C. Is the rex server configured to run rexd?	See D.	Configure the rex server to run rexd by editing /etc/inetd.conf on the rex server, uncommenting the rpc.rexd line, and issuing the inetd -c command. See A.
D. Is there a route to the rex server?	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.	Add a route using the route command. See E.
E. on command works now?	Problem solved.	See F.
F. Does the following error message appear? on: cannot connect to server on <host >	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.	See Flowchart 18.



Flowchart 21: User ID Not Valid

User ID Not Valid (Flowchart 21)

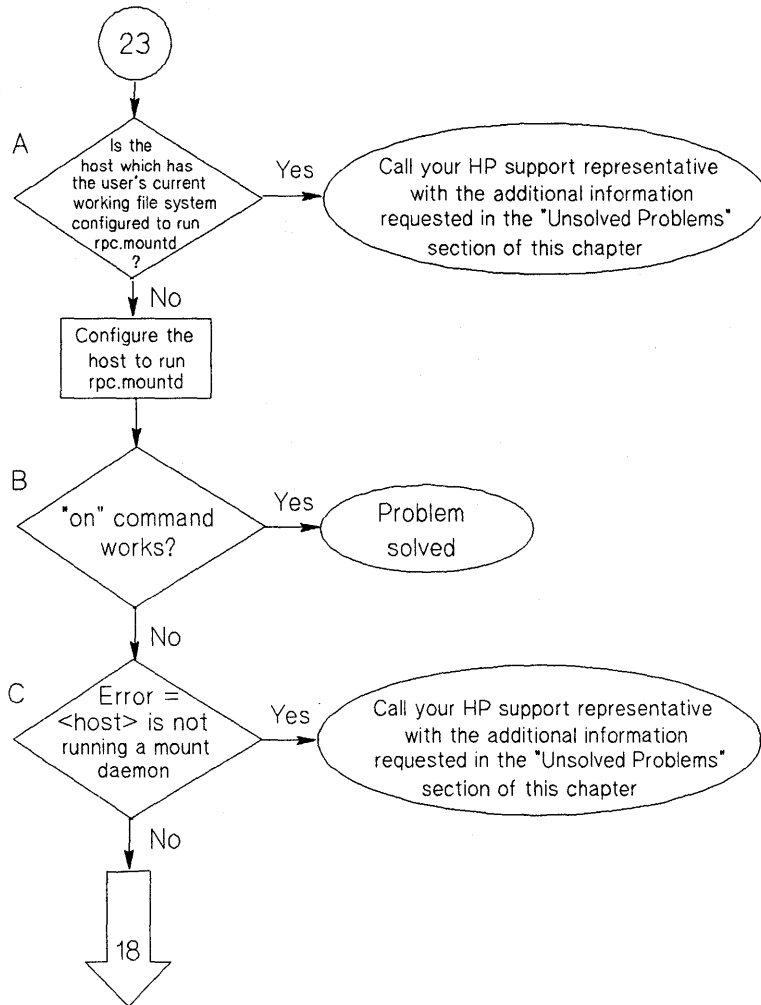
Question	Yes: Action	No: Action
A. Does the user have an account on the rex server with a uid which matches the user's uid on the client?	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.	Create an account on the rex server for the user with a matching uid. See B.
B. on command works now?	Problem solved.	See C.
C. Does the following error message appear? on: rexd: user id is not valid	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.	See Flowchart 18.



Flowchart 22: User ID Denied Access

User ID Denied Access (Flowchart 22)

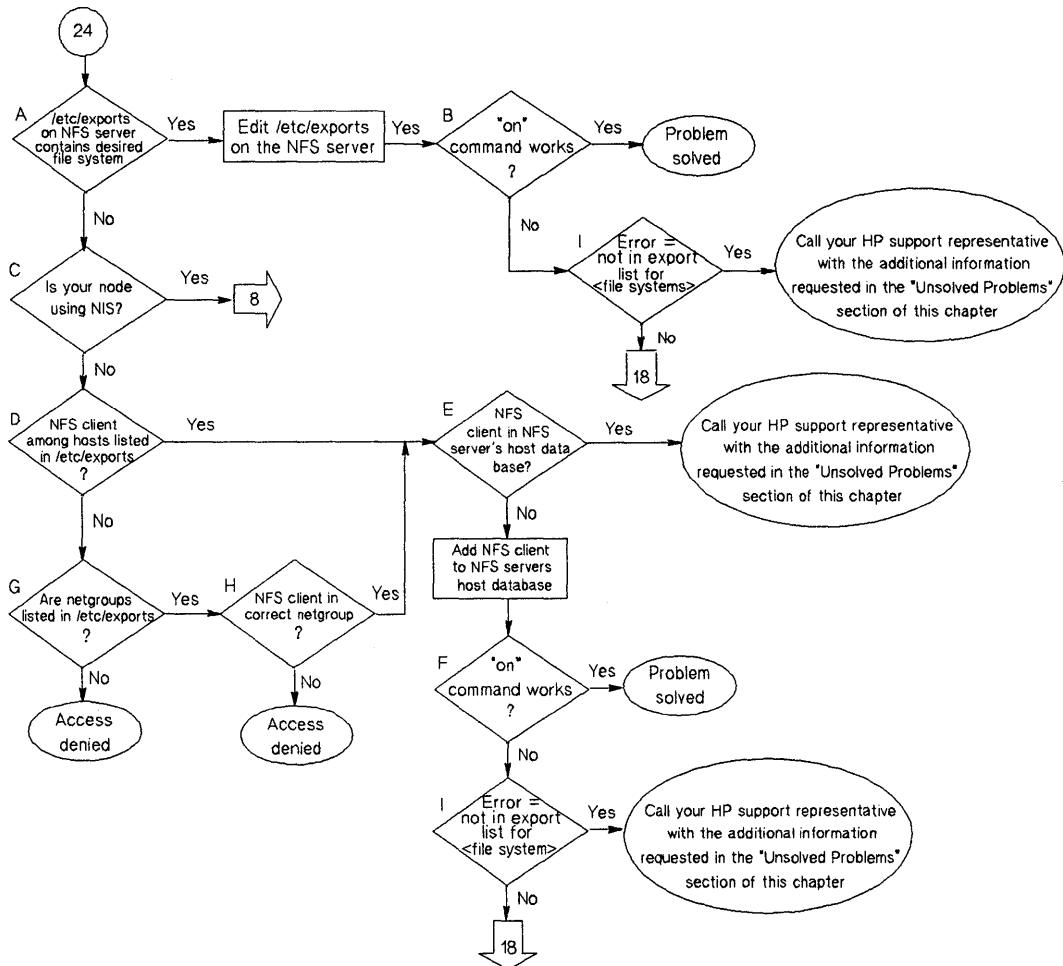
Question	Yes: Action	No: Action
A. Rexd or rex server started with -r option?	See B.	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.
B. User or client host is equivalenced by entry in .rhosts or /etc/hosts.equiv file?	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.	Edit .rhosts or /etc/hosts.equiv file to add an entry for the user or the client host. See C.
C. on command works now?	Problem solved.	See D.
D. Does the following error message appear? on <server>: rexd: user id <uid> denied access	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.	See Flowchart 18.



Flowchart 23: REX Server Not Running Mount Daemon

REX Server Not Running Mount Daemon (Flowchart 23)

Question	Yes: Action	No: Action
A. Is the host which has the user's current working directory physically mounted configured to run rpc.mountd?	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.	Configure the NFS server node to run rpc.mountd by editing its /etc/inetd.conf and executing inetd -c. See B.
B. on command works now?	Problem solved.	See C.
C. Does the following error message appear? on: <server> rexd: <host> is not running a mount daemon	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.	See Flowchart 18.

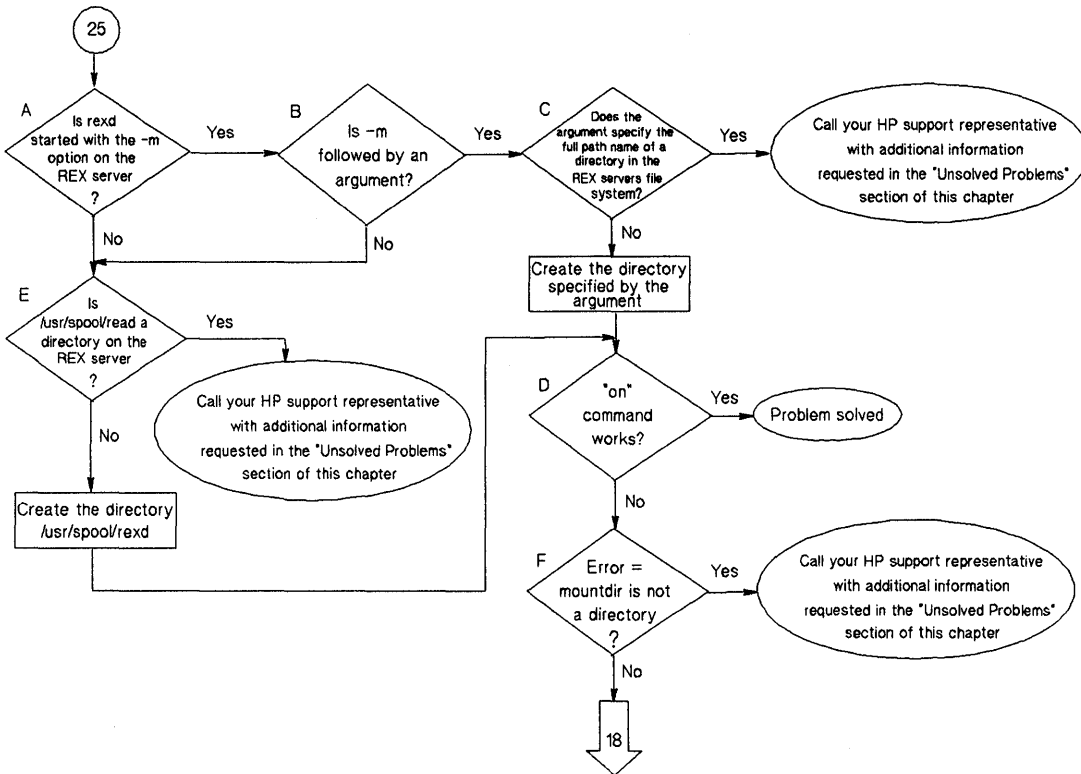


Flowchart 24: REX Server Denied Access Through /etc/exports

REX Server Denied Access through /etc/exports (Flowchart 24)

Question	Yes: Action	No: Action
A. Does /etc/exports on the NFS server contain desired directory?	Edit /etc/exports on the NFS server to contain the directory rather than the directory. See B.	See C.
B. on command works now?	Problem solved.	See I.
C. Is your node using the Network Information Service (NIS)?	See Flowchart 8.	See D.
D. If hosts are listed in the desired /etc/exports entry, is the NFS client one of them?	See E.	See G.
E. Is the NFS client in the NFS server's host database?	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.	Add NFS client to the NFS server's host database. See F.
F. on command works now?	Problem solved.	See I.
G. Are netgroups found for the desired /etc/exports entry?	See H.	Access for this client is denied.
H. Is the client included in a netgroup which is listed in the desired /etc/exports entry?	See E.	Access for this client is denied.

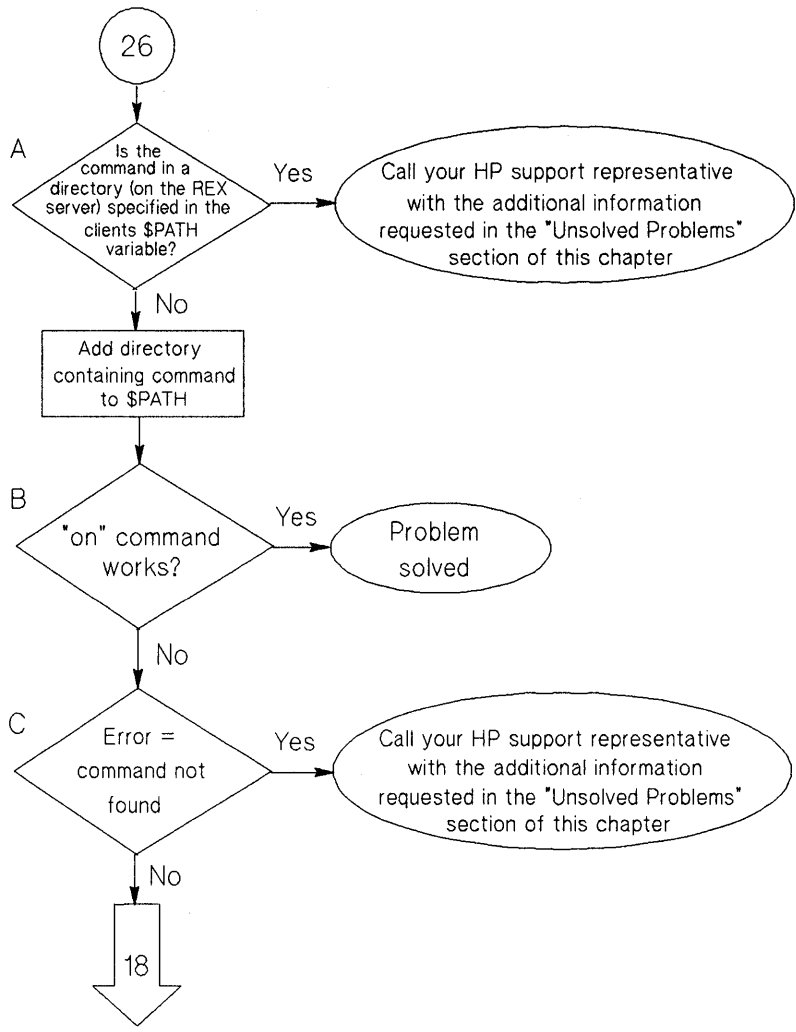
Question	Yes: Action	No: Action
I. Does the following error message appear? on <server>: rexd: not in export list for directory	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.	See Flowchart 18.



Flowchart 25: Mount Point Not a Directory

Mount Point Not a Directory (Flowchart 25)

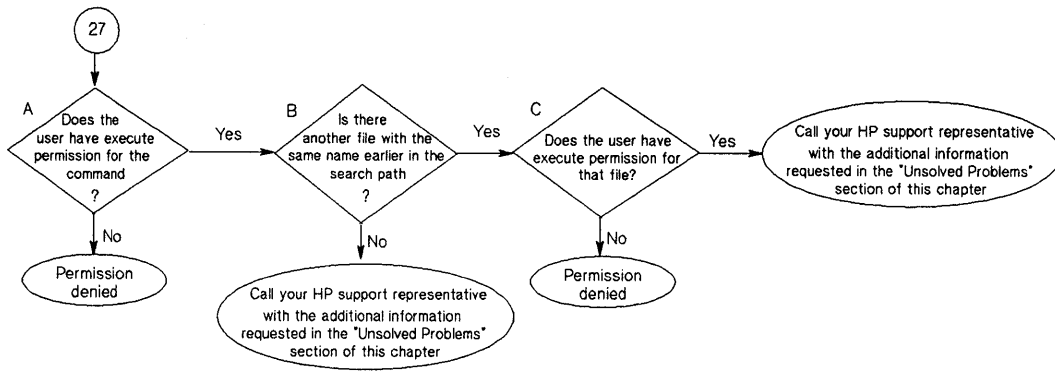
Question	Yes: Action	No: Action
A. Is rexd on the REX server started with the -m option?	See B.	See E.
B. Is -m followed by a full path name?	See C.	See E.
C. Does the full path name specify a directory on the REX server?	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.	Create the directory specified by the path name. See D.
D. on command works now?	Problem solved.	See F.
E. Is /usr/spool/rexd a directory on the REX server?	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.	Create the directory /usr/spool/rexd. See D.
F. Does the following error message appear? on <server>:rexd: (<mountdir>) is not a directory	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.	See Flowchart 18.



Flowchart 26: Command Not Found

Command Not Found (Flowchart 26)

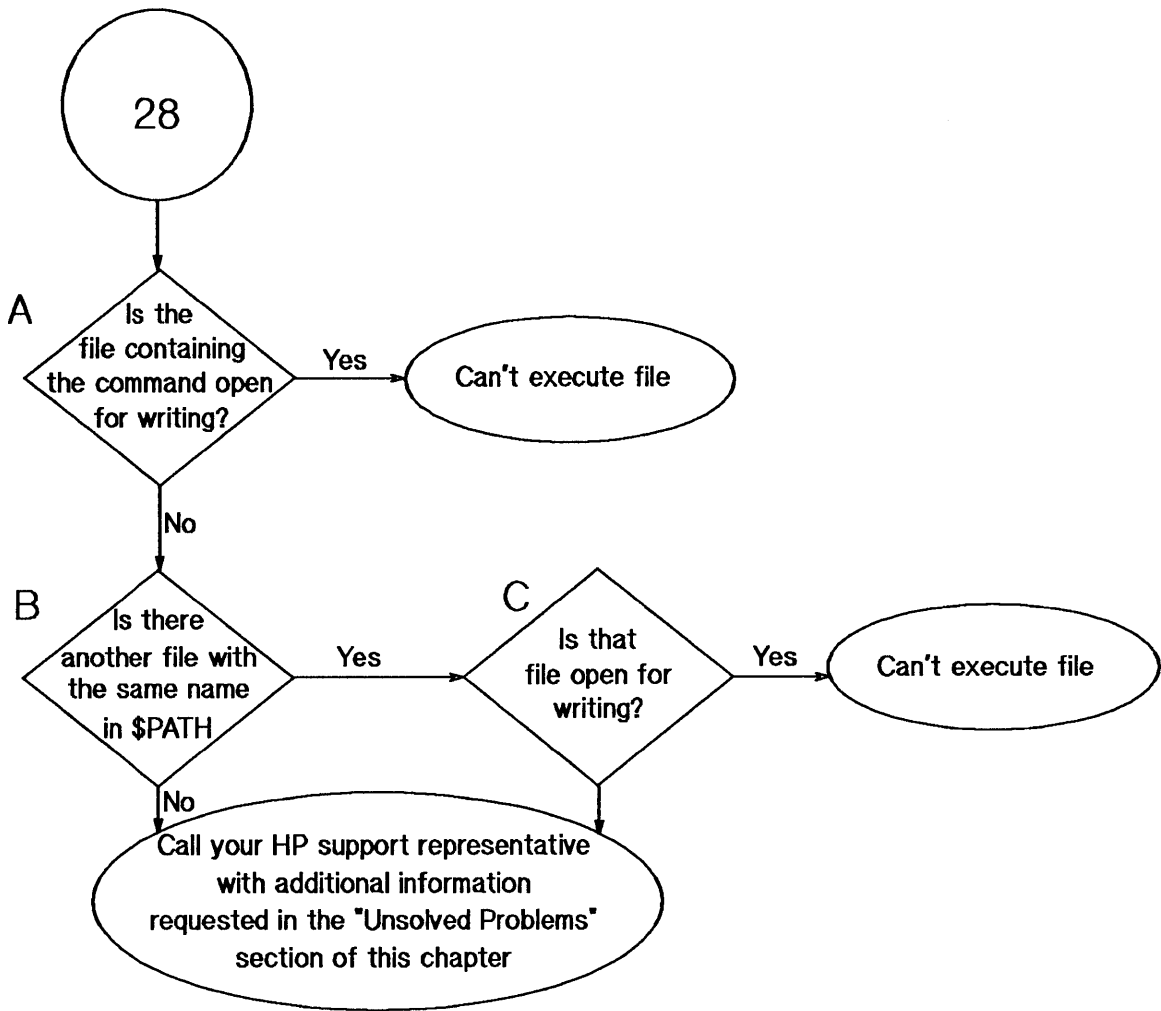
Question	Yes: Action	No: Action
A. Is the command in a directory (visible on the REX server) which is specified in the user's \$PATH variable?	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.	Add the directory containing the command to the user's \$PATH variable. See B.
B. on command works now?	Problem solved.	See C.
C. Does the following error message appear? on: <server> :rexd: command not found	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.	See Flowchart 18.



Flowchart 27: Permission Denied

Permission Denied (Flowchart 27)

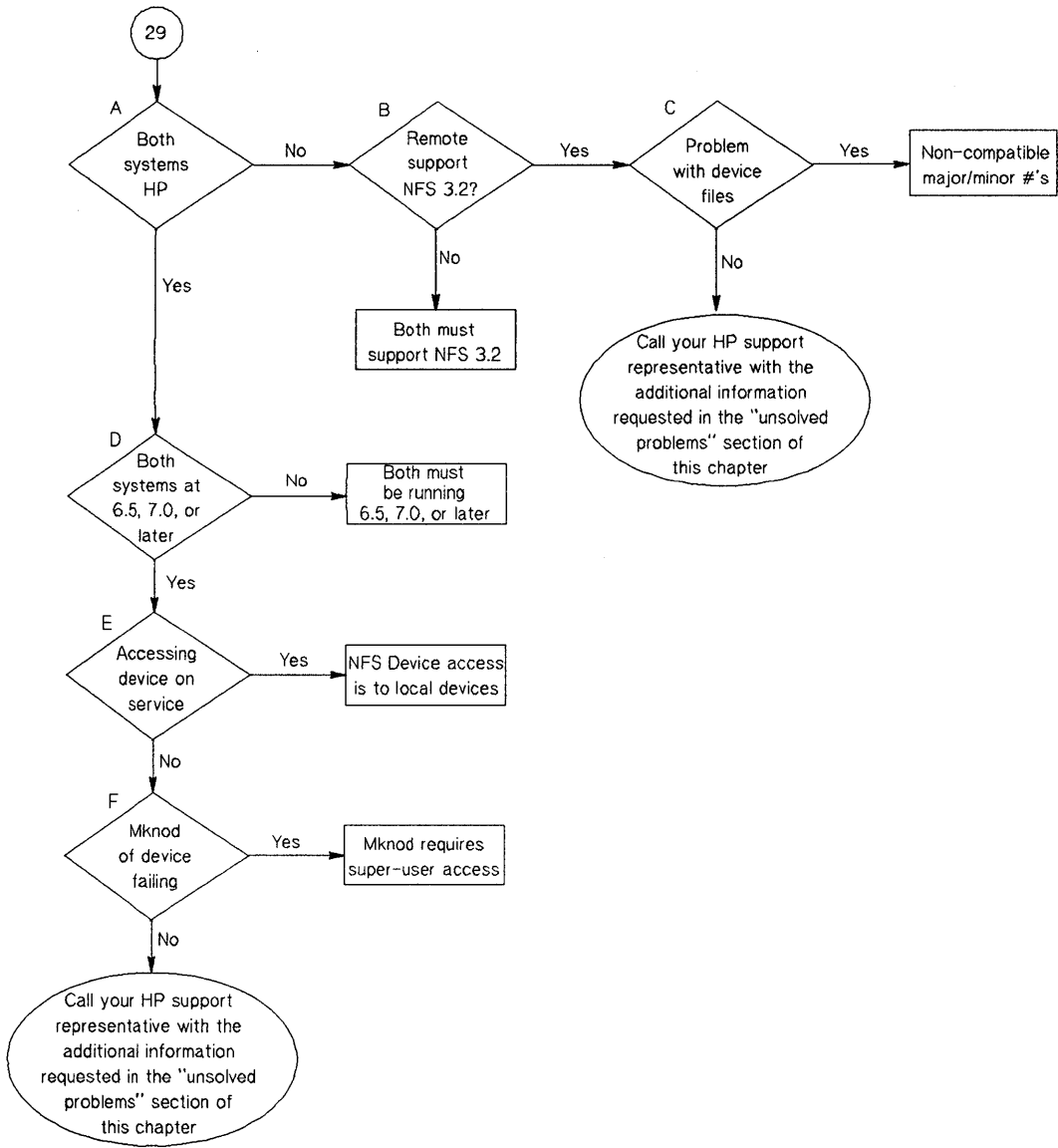
Question	Yes: Action	No: Action
A. Does the user have execute permission for the command?	See B.	Permission denied.
B. Is there another file with the same name in a directory earlier in the user's \$PATH variable?	See C.	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.
C. Does the user have execute permission for that file?	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.	Permission denied.



Flowchart 28: Text File Busy

Text File Busy (Flowchart 28)

Question	Yes: Action	No: Action
A. Is the file containing the command open for writing?	Can't execute file.	See B.
B. Is there another file with the same name in a directory earlier in the user's \$PATH variable?	See C.	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.
C. Is that file currently open for writing?	Can't execute file.	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.



Flowchart 29: Device files/named pipes

Device files/named pipes (Flowchart 29)

Question	Yes: Action	No: Action
A. Are both systems HP systems?	See D.	See B.
B. Does the remote system support NFS 3.2 device files?	See C.	Both systems must support NFS 3.2. Consider mounting with the <code>-nodevs</code> option.
C. Is the problem with accessing device files?	Non-HP systems will have incompatible device major and minor numbers and format. This access will not work.	Problem is with named pipes. Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.
D. Are both systems running HP-UX releases 6.5, 7.0, or later?	See E.	Both systems must be running HP-UX release 6.5, 7.0, or later. Consider upgrading to the latest release of HP-UX.
E. Is the physical device represented by the device file attached to the server?	NFS device file access is to devices local to the client system.	See F.
F. Is an attempt to <code>mknod</code> a device file failing?	Creation of device files requires superuser access. Login as root on the server to create the device file.	Call your HP support representative with the additional information requested in the Unsolved Problems section of this chapter.



HP NFS Services vs. Local HP-UX

If you have applications running on HP-UX, they may behave differently over NFS Services. Use this appendix to understand the basic differences between NFS Services and local HP-UX operations.

HP NFS Services Networking Operation	Local HP-UX Operation
Append Mode	
If two processes operating on different clients open the same file using O_APPEND, the write operation may not append data to the file.	If two processes open the same file using O_APPEND, the write operation should append information to the file.
chacl(1)	
You can only use the -F option. The other options of chacl are not supported over NFS.	You can use all options locally.
Device Files	
NFS does not support remote access to device files, but does support local access to device files via NFS.	HP-UX supports local access to device files.
File Locking	
NFS supports remote file locking for NFS reads and writes in advisory mode only.	HP-UX supports local file locking in advisory and enforcement modes.
getacl(2) system call	

HP NFS Services Networking Operation	Local HP-UX Operation
Is not supported over NFS.	Is supported locally.
Group Membership	
A user may be a member of 16 groups. If a user who is a member of more than 16 groups attempts to access a file, the system accesses only the first eight groups for permission checking.	A user may be a member of up to 20 groups.
lseek(2)	
If two processes operating on different clients write to the same file, lseek with whence = SEEK_END may not set the file pointer to the desired location.	If two processes write to the same file, lseek with whence = SEEK_END should set the file pointer to the desired location.
mknod(1M) Command	
The mknod command will work only with named pipes over NFS.	You can use the mknod command locally for all file types.
Mount Points	
When operating in an HP-UX cluster environment, only directories or files mounted on the cluster root server can contain mount points for NFS mounts. File systems mounted on cluster auxiliary servers cannot contain NFS mount points.	NFS mount points can exist on any mounted directory.
Named Pipes	
NFS named pipes cannot be used to communicate between machines in the same diskless cluster.	Named pipes can be used to communicate among clients in a diskless cluster.
Reading Directories	

HP NFS Services Networking Operation	Local HP-UX Operation
You cannot use the read call to read a remote directory, rather you should use readdir .	You can use the read call to read a local directory. However, to do so can restrict migration of programs to future HP-UX versions.
setacl(2) system call	
Is not supported over NFS.	Is supported locally.
setaclentry(3) library routine	
Is not supported over NFS.	Is supported locally.
Superuser Permission	
<p>The superuser UID 0 is mapped to -2 by default.</p> <p>Anything requiring superuser permission may not work over NFS. For example, a superuser may not be able to perform the following tasks:</p> <ul style="list-style-type: none"> - Link and unlink directories. - Alter directories such as /, /etc, and /bin. - Use chmod to set sticky or setuid bits. - Do a mknod of device files. 	Superuser has permission to perform any operation locally (by definition).
System Time	

HP NFS Services Networking Operation	Local HP-UX Operation
<p>Commands that access clocks on different systems may not provide consistent times since system clocks differ.</p> <p>For example, if you give the <code>utime</code> command a NULL pointer for the times value, the following process occurs:</p> <ol style="list-style-type: none"> 1. The system sets the access time and modification time according to the client node clock. 2. It then sends these times over to the server which changes the inode to reflect the new access and modification times. 3. The server node identifies the change in the inode and thus, modifies the inode's status change time according to its own clock. <p>The result is a high probability of differing times between the server's access and modification times versus its status change time.</p> <p><i>Note:</i> If operating in an HP-UX cluster environment, all nodes in the cluster have the same time as the root server's clock. Therefore, clock skew problems exist only if the root server's clock is different from other NFS servers.</p>	<p>Commands that access clocks on the local system provide consistent times.</p>
<p>Unlinking</p>	

HP NFS Services Networking Operation	Local HP-UX Operation
<p>The server does not keep state information and does not know if a process has a file open. See the following explanation:</p> <ul style="list-style-type: none"> - The server will unlink a file if it receives a request to do so; thus, subsequent requests for the file will result in an error. - If a process opens a file and then unlinks it, the client renames the file so it appears to be gone. When the process quits, the client then unlinks the renamed file. - If the unlink request comes from a different node than from where the open request came from, the file is deleted. 	<p>If you open a local file and unlink it before you close the file, the file descriptor for the open file will still be valid to access the file.</p>
<p>yppasswd(1) Command vs. passwd(1) Command</p>	
<p>This command does have a <i>password aging</i> feature.</p> <p>The superuser must know the current password to change another user's password. The password must contain:</p> <ul style="list-style-type: none"> - At least five characters if it includes special characters and any combination of lowercase letters, uppercase letters, and numbers. Also, any combination that includes numbers, lowercase letters, and uppercase letters. 	<p>This command has a <i>password aging</i> feature.</p> <p>Superuser does not have to know the password to change another user's password. The following rules apply to the password:</p> <ul style="list-style-type: none"> - Each password must have six or more characters: at least two alpha characters and at least one numeric or special character.
<p>yppasswd(1) Command vs. passwd(1) Command (continued)</p>	

HP NFS Services Networking Operation	Local HP-UX Operation
<ul style="list-style-type: none"> - At least six characters if it includes lowercase letters and numbers, uppercase letters and numbers, or a combination of lowercase and uppercase letters. - At least seven characters if it includes all lowercase letters or all uppercase letters. 	<ul style="list-style-type: none"> - Each password must differ from the user's login name and any reverse or circular shift of that name. - New passwords must differ from the old by at least three characters.
pathconf/fpathconf	
<p>The following variables for the pathconf/fpathconf system calls are not supported over NFS:</p> <ul style="list-style-type: none"> <code>_PC_CHOWN_RESTRICTED</code> variable <code>_PC_LINK_MAX</code> variable <code>_PC_NAME_MAX</code> variable <code>_PC_NO_TRUNC</code> variable <code>_PC_PATH_MAX</code> variable <p>The following variables for the pathconf/fpathconf system calls return local information over NFS:</p> <ul style="list-style-type: none"> <code>_PC_MAX_CANON</code> variable <code>_PC_MAX_INPUT</code> variable <code>_PC_VDISABLE</code> variable <p>The following variable for the pathconf/fpathconf systems calls is supported over NFS:</p> <ul style="list-style-type: none"> <code>_PC_PIPE_BUF</code> variable 	<p>All variables are supported locally for the pathconf/fpathconf system calls:</p> <ul style="list-style-type: none"> <code>_PC_CHOWN_RESTRICTED</code> variable <code>_PC_LINK_MAX</code> variable <code>_PC_NAME_MAX</code> variable <code>_PC_NO_TRUNC</code> variable <code>_PC_PATH_MAX</code> variable <ul style="list-style-type: none"> <code>_PC_MAX_CANON</code> variable <code>_PC_MAX_INPUT</code> variable <code>_PC_VDISABLE</code> variable <ul style="list-style-type: none"> <code>_PC_PIPE_BUF</code> variable

Moving from RFA to NFS

Remote File Access (RFA), one of the Network Services, has been discontinued. When you used networks consisting of all HP systems, RFA provided distributed file access among HP 9000 computers. In order to maintain distributed file access, you must move to NFS Services.

Why Move to NFS Services?

Using NFS Services in place of the RFA service has several advantages:

- NFS works with other vendors' equipment and other operating systems.
- NFS is a defacto industry standard.
- NFS allows transparent file access.
- NFS with the Network Information Service (NIS) provides centrally administered databases.

Use this appendix to translate your RFA applications to NFS applications.

Similarities

HP NFS Services and RFA have the following similarities:

- No remote device access.
- Not all UNIX semantics are fully supported.

Differences

Refer to the following table for a list of differences between HP NFS and RFA.

NFS Services	RFA (Discontinued)
You can run <code>setuid</code> programs accessing data on remote directories or files.	You cannot run <code>setuid</code> programs accessing data on remote directories or files.
NFS operates in a heterogeneous operating system environment.	RFA operates on HP-UX operating systems only.
Only the superuser can perform remote NFS mounts.	All users can establish access to remote directories or files.
You can centrally administer your databases using NIS.	You have no centrally administered database.
All users with read access to the mount point can read the remote directory.	Only users performing <code>netunam</code> can access the remote directories or files.
Read and write file caching occurs on the clients; read caching occurs on the servers.	Read and write file caching occurs on the servers; caching does not occur on the clients.
The servers are stateless (do not remember client activities) and therefore, can be rebooted without interfering with client activities. (The client can resume access to the server when it is rebooted.)	The servers have state and therefore, remember the activities in which the client is involved.
One mount gives you access to only one directory.	One <code>netunam</code> gives you access to all directories or files under the root directory.

Changing Scripts from RFA to NFS

Changing RFA scripts to NFS requires only minor changes. You can change both shell scripts that accept different path names and those that use hard-coded path names.

Shell Scripts that Accept Different Paths

Shell scripts that accept different paths require the following modifications:

- You must perform a remote mount of a directory or directory in *one* of the following ways:
 - As part of the script.
 - Before executing the script.

Since superuser must execute mounts, the script must be setuid root if the mount is performed as part of the script.

Caution Having setuid root scripts is a potential security problem.

If the script's owner does not have superuser permissions, the superuser can configure `/etc/checklist` to automatically mount the remote directories or files at boot time. This process allows users to execute scripts without checking to see if the remote directory is accessible.

- Remove all calls to `netunam` from the script. Removing these calls prevents `netunam` failures from causing the scripts to fail.

Shell Scripts with Hard-Coded Paths

You can handle shell scripts with hard-coded path names in two ways:

- Change the path name in the script to correspond to the NFS mount point.
- Create a path name for the NFS mount point which corresponds to the path name in the script.

To mount the remote directory either as part of the script or automatically via `/etc/checklist`, you must modify the shell scripts as described in the previous section, `Shell Scripts that Accept Different Paths`.

Change Directorys

Change the path name in the script to correspond to the NFS mount point.

EXAMPLE: The script has a hard-coded path name of `/net/systemB/project`. Mount the remote directory `/project` on `/user/project` as follows:

```
mount systemB:/project /user/project
```

Now change the script to use the path name `/user/project` in place of `/net/systemB/project`.

Create New Directories

Create a path name for the NFS mount point that corresponds to the path name in the script.

EXAMPLE: The script has a hard-coded path name of `/net/systemB/project` which accesses the remote directory `/project`. To keep the path name the same:

1. Remove the network special file `/net/systemB`.
2. Create the directories `/net/systemB` and `/net/systemB/project`:

```
mount systemB:/project /net/systemB/project
```

Note For RFA, access to the remote system occurred via a network special file. Creating an NFS mount point with the same name as the network special file for the remote system could cause confusion. Problems will not occur if you remove the network special file.

All remote access will then be via mount points that have the same names as the network special files that were removed.



NFS in an HP-UX Cluster Environment

Reference this appendix for interactions between NFS Services and HP-UX cluster environments using diskless capabilities.

HP-UX Cluster Terms

Term	Definition
Context Dependent File (CDF)	A hidden directory which contains all the versions of a file or directory needed by the different cnodes.
Cluster	One or more workstations linked together with a local area network (LAN), but consisting of only one root directory. For more information on cluster concepts, see <i>Managing Clusters of HP9000 Computers: Sharing the HP-UX Filing System</i> .
Cluster Auxiliary Server	A cluster client with a disk drive that contains files shared by the other members of the cluster.
Cluster Client	A node in an HP-UX cluster that uses networking capabilities to share directories or files, but does not have its root directory directly attached. For HP-UX 8.0, cluster clients can have locally mounted disks for local data storage.
Cluster Node (Cnode)	Any node operating in an HP-UX cluster environment, including cluster clients and cluster servers.
Cluster Root Server	The only node in an HP-UX cluster that has the root directory directly attached to it.
Homogeneous Cluster	A diskless cluster composed of nodes of only one computer architecture (e.g., Series 300/400 only).
Mixed Cluster	Diskless cluster consisting of cnodes of multiple architectures.

NFS Configuration and Maintenance

Configure

If you configure NFS on the cluster root server, you must also configure NFS on all clients in the cluster. If the cluster root server does not have NFS configured, then none of the clients can use NFS.

Daemons

- The `nfsd` daemon should be running on the cluster root server and all cluster auxiliary servers if it is servicing NFS requests. Any `nfsd` daemon running on any other cnode is ignored.
- The `rpc.mountd` daemon should be running on the cluster root server and all cluster auxiliary servers if servicing NFS requests. Any `rpc.mountd` daemon running on any other cnode is ignored.
- The `biod` daemon should be running on all cnodes in the cluster.

Mount/Unmount

- If a cnode mounts a remote directory, all cnodes in the cluster can access the remote directory.
- If using NFS to mount a directory attached to a cluster, you must use the host name of the directory server as the node name specified in the `mount` command.
- If a cnode mounts a remote directory, any cnode in that cluster can unmount the remote directory.
- If a cnode unmounts a directory, all cnodes in the cluster will have that directory unmounted.
- Clients should not execute `umount -a`.
- NFS mount points may not exist on directories or files mounted on cluster auxiliary servers.

Context Dependent Files (CDF)

When accessing a *CDF* via an NFS mount, the *CDF* member is chosen based on the context of the NFS server, not the accessing node. Since this access method may return unexpected results, HP recommends you do not mix *CDF*s with NFS.

Clock Skew

All nodes in the HP-UX cluster have the same time as the cluster root server's clock. Therefore, clock skew problems exist only if the cluster root server's clock is different from other NFS servers.

NIS Configuration and Maintenance

HP recommends that you execute `ypserv` only on the cluster root server for the two following reasons:

- For better performance.
- For assurance that the cluster root server is the only Network Information Service (NIS) server for that cluster.

Troubleshooting

You can troubleshoot NFS specific problems from the cluster root server and cluster auxiliary server as follows:

- If you are trying to mount an NFS directory, ensure you are using the cluster root server's host name as the node specified in the `mount` command.
- If problems exist in the link, cnodes will not be able to boot. Since link diagnostics reside on the root disk, first test the link from the cluster root server. (Refer to the *Installing and Administering LAN*, *Installing and Administering FDDI/9000 Software*, or *Installing and Administering Token Ring/9000* manuals.)

Password Security

This appendix explains the restrictions and limitations on the use of encrypted passwords and the secure password file with the Network Information Service (NIS). If you wish to review the normal use of passwords with the Network Information Service, see the NIS Configuration and Maintenance section in this manual. If you require additional information on the secure password file, see `passwd` in the *HP-UX Reference* manual.

The HP 9000 now supports a secure password file (`/.secure/etc/passwd`) used to hide your encrypted passwords from non-privileged users. Therefore, it is probable that if you use the secure password file, your `/etc/passwd` file will probably contain (in the password field) a character that is not part of the set of characters used in an encrypted password (e.g. `*`). The NIS database will not contain encrypted passwords if you use this `/etc/passwd` file to build your NIS password database. This prevents non-privileged users from reading your passwords, because anyone with access to NIS commands such as `ypcat` or NIS library routines such as `yp_first` and `yp_next` can read the NIS database.

If you are using the secure password file only to use the auditing subsystem and you do not need to hide your encrypted passwords, you can maintain an `/etc/passwd` file that contains encrypted passwords that match those in your secure password file. You can then use this `/etc/passwd` file to build your NIS database.

Note A password in the `/.secure/etc/passwd` file takes precedence over the password stored in NIS.

If you wish to hide the encrypted passwords in your HP systems and wish to continue to use the NIS password database to maintain other information kept on the password file, you can do the following:

- Build your NIS password database on the HP NIS master server using a password file that does not contain encrypted passwords (e.g. uses "*" in the password field).
- On an HP NIS client, maintain a copy of the secure password file so the passwords in that file will be used at login.
- On an HP or non-HP NIS client, maintain the encrypted password in the `/etc/passwd` file through an NIS escape.

EXAMPLE:

```
+ username:encrypted_passwd::::
```

Glossary

A

Alias A term for referencing alternate networks, hosts, and protocols names.

ARPA Advanced Research Projects Agency. A U.S. government agency that was instrumental in developing and using the original ARPA Services networking standards.

B

Bind A process by which a client locates and directs all requests for data to a specific server. A process of establishing the address of a socket that allows other sockets to connect to it or to send data to it. An acronym for Berkeley Internet Name Domain. The BIND Name Server is a distributed network lookup service.

C

CDF (Context Dependent File) A hidden directory that contains all the versions of a file needed by the different cnodes.

Client A node that requests data or services from other nodes (servers). A process that requests other processes to perform operations. *Note:* An NFS client can also be configured as any combination of an NFS server, NIS client, or NIS server. (An NIS server must also be configured as an NIS client.)

Clock Skew A difference in clock times between systems.

Cluster One or more workstations linked together with a local area network (LAN), but consisting of only one root file system.

Cluster Auxiliary Server A cluster client with a disk drive that contains files shared by the other members of the cluster.

Cluster Client A node in an HP-UX cluster that uses networking capabilities to share file systems,, but does not have its root file system directly attached. For HP-UX 8.0, cluster clients can have locally mounted disks for local data storage.

Cluster Root Server The only node in an HP-UX cluster that has the root file system directly attached to it.

Cnode (Cluster Node) Any node operating in an HP-UX cluster environment, including cluster clients, cluster auxiliary servers, and the cluster root server.

D

Daemon Background programs that are always running, waiting for a request to perform a task.

E

Escape Sequence (NIS) Characters used within files to force inclusion and exclusion of data from NIS databases. The escape sequences are as follows.

- * + (plus sign)
- * - (minus sign)
- * +@netgroup_name
- * -@netgroup_name

Export To make a file system available to remote nodes via NFS.

External Data Representation (XDR) A protocol that translates machine-dependent data formats (i.e., internal representations) to a universal format used by other network hosts using XDR.

F

File System A directory structure used to organize files.

G

GID A value that identifies a group in HP-UX.

Global (NIS) A means of access in which the system always reads NIS maps rather than the local ASCII files.

H

Hard Mount A mount that causes NFS to retry a remote file system request until it succeeds, you interrupt it (default option), or you reboot the system.

Home Node A term used in Virtual Home Environment (VHE) to refer to the machine on which a user's home directory physically resides.

Host A node that has primary functions other than switching data for the network.

Host Node A term used in Virtual Home Environment (VHE) to refer to the node a user is logged in to. This node environment is set up from the configuration files found on the user's home node.

I

Import To obtain access to a remote file system from an outside source; to mount.

Internet Address A four-byte quantity that is distinct from a link-level address and is the network address of a computer node. This address identifies both the specific network and the specific host on the network.

Interruptable Mount A mount that allows you to interrupt an NFS request by pressing an interrupt key. (Though the interrupt key is not standardized, common ones include [CTRL] - [C] and [BREAK].)

K

Key (NIS) A string of characters (no imbedded blanks or tabs) that indexes the values within an NIS map so the system can easily retrieve information. For example, in the passwd.byname map, the users' login names are the keys and the matching lines from /etc/passwd are the values.

L

Local (NIS) A means of access in which the system first reads the local ASCII file. If it encounters an escape sequence, it then accesses the NIS databases.

M

Map (NIS) A file consisting of logical records; a search key and related value form each record. NIS clients can request the value associated with any key within a map. NIS map is synonymous with NIS database.

Map Nickname (NIS) A synonym for the NIS map name when using certain NIS commands.

Master Server (NIS) The node on which one or more NIS maps are constructed from ASCII files. These maps are then copied to the NIS slave servers for the NIS clients to access.

Mount To obtain access to a remote or local file system or directory (import).

Mount Point The name of the directory on which a file system is mounted.

N

Netgroup A network-wide group of nodes and users defined in `/etc/netgroup`.

Network Information Service (NIS) An optional network service composed of databases (maps) and processes that provide NIS clients access to the maps. The NIS service enables you to administer these databases from one node. NIS may or may not be active; check with your system administrator.

Network Lock Manager A facility for locking files and synchronizing access to shared files.

Network Status Monitor A daemon running on all network computers to maintain stateful locking service within NFS. It also allows applications to monitor the status of other computers.

NFS Network File System.

NIS Client A node that requests data or services from NIS servers. An NIS process that requests other NIS processes to perform operations. *Note:* An NIS client can also be configured as any combination of an NIS server, NFS client, or NFS server. (An NIS server must also be configured as an NIS client.)

NIS Database See Map (NIS).

NIS Domain A logical grouping of NIS maps (databases) stored in one location. NIS domains are specific to the NIS network service and are not associated with other network domains.

NIS Map See Map (NIS).

NIS Password The password for a user's login ID that exists in the NIS passwd map. The NIS password is the same one as the user password, but is administered through the NIS. You do not have to have an NIS password to access the NIS databases.

NIS Server A node that provides data (maps) or services to other nodes (NIS clients) on the network using NIS. An NIS process that performs operations as requested by other NIS processes. *Note:* An NIS server must also be configured as an NIS client. It can also be configured as an NFS server, NFS client, or both.

Node A computer system that is attached to or is part of a computer network.

P

Propagate To copy maps (data) from one NIS server to another.

Protocol The rules and steps by which servers and clients exchange data and control information.

R

Remote Execution Facility (REX) A facility which allows a user to execute commands on a remote node.

Remote Procedure Call (RPC) A call made by clients either to access server information or to request action from servers.

Remote Procedure Call Protocol Compiler (RPCGEN) A remote procedure call compiler used to help programmers write RPC applications by automatically generating necessary programs and code fragments.

S

Server A node that provides data or services to other nodes (clients) on the network. A process that performs operations as requested by other processes. *Note:* An NFS server can

also be configured as any combination of an NFS client, NIS client, or NIS server. (An NIS server must also be configured as an NIS client.)

Slave Server (NIS) A node that copies NIS maps from the NIS master server and then provides NIS clients access to these maps.

Soft Mount An optional mount that causes access to remote file systems to abort requests after one NFS attempt.

Stateless Server Servers do not maintain (preserve) information relating to each file being served. Each file request moves across the network with the parameters attached to it locally (e.g., read and write privileges).

Steady State Servers maintain (preserve) information relating to each file being served. For NIS, the information contained in an NIS map is consistent among all NIS servers within a given NIS domain (i.e., is not in the process of being updated).

U

UID A value that identifies a user in HP-UX.

Unmount To remove access rights to a file system or disk that was mounted via the mount command.

update The HP-UX command that installs software onto the system.

V

Value (NIS) A unit of information stored in NIS maps; each value has a corresponding key (index) so the system can easily retrieve it. For example, in the `passwd.byname` map, the users' login names are the keys and the matching lines from `/etc/passwd` are the values.

VHE See Virtual Home Environment.

Virtual Home Environment (VHE) A network service that allows users to log in at host nodes and utilize their home nodes' execution environments.

X

XDR See External Data Representation.

Index

!

- \$HOME/.rhosts, 7-12
- /etc/hosts.equiv, 7-12
- /tmp_mnt, 2-26

A

- Adding computers, to a network, 3-9
- Automatic mounts, 4-53
- automount, 2-25, 9-1
 - advanced automount features, 9-28
 - automount concepts, 9-2
 - automount error messages, 9-21
 - automount maps, 9-2
 - command line options, 9-16
 - configuration checklist, 9-6
 - create direct and indirect maps, 9-10
 - create master map, 9-8
 - creating NIS maps, 9-13
 - direct maps, 9-4, 9-12
 - hierarchical mounts, 9-28
 - hosts map, 9-4
 - indirect maps, 9-3, 9-10
 - integrate with NIS, 9-13
 - modifying automount maps, 9-18
 - modifying direct maps, 9-18
 - modifying indirect maps, 9-18
 - modifying the master map, 9-18
 - NFS configuration, 9-7
 - planning and design, 9-7
 - replicated servers, 9-28
 - shutting down automount, 9-19
 - special automount characters, 9-11
 - subdirectory notation, 9-29
 - verify automount configuration, 9-18

B

- Binding, client to server defined, 2-1
- biod daemon
 - Defined, 4-8

C

Clients

- Creating NFS clients, using manual method, 4-44
- Creating NFS clients, using SAM, 4-14–4-15
- Defined, for NFS, 2-1
- NIS, 2-17, 10-17
- Unmounting directories or files, manually, 4-61

Clock skews, 4-66

- Clusters, C-3

Clusters

- CDFs, C-1–C-2
- Clock skews, C-3
- Daemons, C-2
- Defined, C-1
- HP-UX, C-1
- Mounts, C-2
- NFS configuration, C-2
- NFS maintenance, C-2
- NIS configuration, C-3
- NIS maintenance, C-3
- Troubleshooting, C-3
- Unmounts, C-2

Commands

- Common, 10-1
- domainname, 7-14, 10-18
- makedbm, 7-14
- NFS, commonly used, 10-5
- NIS, 7-14
- NIS, commonly used, 10-18
- on, 10-14
- see also:* on command
- on, command, 5-2
- rpcinfo, 10-8
- rup, 10-10
- rusers, 10-11
- showmount, 10-13
- ybind, 7-14
- ypcat, 7-14, 10-19

- ypinit, 7-14
- ypmatch, 7-14, 10-20
- yppasswd, 7-15, 10-21
- yppasswdd, 7-15
- yppoll, 7-15
- ypserv, 7-15
- ypwhich, 7-15, 10-23
- ypxfr, 7-15

Configuration

- Kernel, 3-8
- Key terms, 4-2, 10-2
- see* Network Information Service
- NFS client using SAM, 4-17
- see* NFS Configuration
- NFS in an HP-UX cluster environment, 3-8, 4-20, 4-48, C-1
- NFS Server using SAM:, 4-19
- see* Virtual Home Environment (VHE)

Configuration files

- /etc/checklist, defined, 4-6
- /etc/checklist, manually edit, 4-53
- /etc/exports, defined, 4-6
- /etc/exports, manually edit, 4-34
- /etc/hosts file, manually edit, 4-27
- /etc/inetd.conf, configuring rexd, 5-6
- /etc/inetd.conf, defined, 4-6
- /etc/inetd.conf, manually edit, 4-23
- /etc/netgroup file, manually edit, 4-32, 7-9
- /etc/netgroup, defined, 4-6
- /etc/netnfsrc, defined, 4-6
- /etc/netnfsrc, manually edit, 4-20, 4-44
- /etc/netnfsrc2, defined, 4-6
- /etc/rpc, defined, 4-7
- /usr/adm/inetd.sec file, manually edit, 4-25
- /usr/adm/inetd.sec, defined, 4-7
- Used for NFS, 4-6

Conventions, used in this manual, 1-5

crontab, 7-30

D

Daemons

- biod, defined, 4-8
- Clusters, C-2
- inetd, configuring rexd, 5-6
- inetd, defined, 4-8

inetd, security for RPC services, 4-23

nfsd, defined, 4-8

pcnfsd, defined, 4-8

portmap, defined, 4-9

Device files

Defined, 2-7

NFS Services vs. Local HP-UX, A-1

Diagnostics, for REX, 5-10

Directories or Files

Preventing access, 4-63

Documentation

Contents of manual, 1-2

Conventions, used in this manual, 1-5

Guide of other services, 1-6

Military Standards, address for obtaining, 1-7

Overview, 1-1

RFC (Request for Comment) documents, address for obtaining, 1-7

domainname command, 7-14, 10-18

Domains, NIS, 10-17

E

Environment simulation, in REX, 5-5

Error messages

on command, 5-10

rexd daemon, 5-11

Troubleshooting network problems, 11-12

Escape sequences, use in NIS databases, 7-8

/etc/checklist file

Defined, 4-6

/etc/checklist file

Manually edit, 4-53

/etc/exports file

Defined, 4-6

/etc/exports file

Manually edit, 4-34

/etc/hosts file

Manually edit, 4-27

/etc/inetd.conf file

Configuring rexd, 5-6

Defined, 4-6

/etc/inetd.conf file

Manually edit, 4-23

/etc/netgroup file

Defined, 4-6

- /etc/netgroup file
 - Manually edit, 4-32, 7-9
- /etc/netnfsrc file
 - Defined, 4-6
- /etc/netnfsrc file
 - Manually edit, 4-20, 4-44
- /etc/netnfsrc2 file
 - Defined, 4-6
- /etc/newconfig file, comparing to existing files, 4-11
- /etc/rpc file
 - Defined, 4-7
- Export directories or files, 4-34
- exportfs, 4-41
- External Data Representation (XDR), 2-12

F

- fcntl(), mapping user calls, 6-1
- File access
 - Defined, for NFS remote, 2-4
 - Device files, defined, 2-7
 - Limiting, 4-32, 4-34
 - Migration of NS to NFS, B-1
 - mknod, creating named pipes, 2-7
 - Named pipes, defined, 2-6
 - Preventing, 4-61
 - Procedure, for NFS remote, 10-6
- File locking, using Network Lock Manager, 6-1
- File systems
 - Automatic mounts, 4-53
 - Availability of, 4-32
 - Manual mounts, 4-56
 - Mounting of, manually, 4-47
 - Preventing access, 4-61
 - Unmounting of, manually, 4-61
- Files
 - NFS configuration, 4-6

G

- GIDs, Setting of, 4-12
- Global maps, for NIS, 7-7

H

Hard mounts

- Defined, 4-47

- hard, option, 4-50

- via /etc/checklist, 4-53

- via mount command, 4-56

- HP-UX clusters, C-1

I

inetd daemon

- Configuring rexd, 5-6

- Defined, 4-8

- Security, for RPC services, 4-23

Installation

- Adding computers, 3-9

- Configure a new kernel, 3-8

- Introduction, 3-1

- Key terms, 3-1

- Preparing the system, 3-5

- Software, 3-6

- Steps to follow, 3-3

- Using the update command, 3-6

K

Key terms, 3-1

- Configuration, 4-2, 10-2

L

- Local maps, for NIS, 7-7

- lockf(), mapping.user calls, 6-1

- Locking protocol, 6-4

- Log files, for NIS, 7-43

M

Maintenance

- see* Network Information Service

- see also:* NFS Maintenance

- NFS Services, 4-59

- VHE, 8-12

makedbm command, 7-14

Maps

Global, for NIS, 7-7

Local, for NIS, 7-7

NIS, maintenance of, 7-35

NIS, modifying, 7-35

NIS, modifying manually, 7-36

Non-standard, 7-36, 7-44

Propagation, 7-30

Maps, NIS, 2-17, 10-16

Master server

Changing of, 7-40

Master servers

NIS, automatic start, 7-20

NIS, configuration, 7-17

NIS, manual start, 7-20

NIS, security, 7-18

Memory, 4-5

mknod command

Creating named pipes, 2-7

NFS Services vs. Local HP-UX, A-2

mount command, executing for manual mount, 4-56

Mount defaults

Defined, 4-50

devs, 4-50

fg, 4-50

hard, 4-50

int, 4-50

port, 4-51

retrans, 4-51

retry, 4-51

rsize, 4-51

rw, 4-51

setuid, 4-52

timeo, 4-52

wsizer, 4-52

mount information, 4-42

Mount options

acdirmax = n, 4-50

acdirmino = n, 4-50

acregmax = n, 4-50

acregmin = n, 4-50

actimeo = n, 4-50

bg, 4-50

devs, 4-50

- fg, 4-50
- hard, 4-50
- int, 4-50
- noac, 4-50
- noauto, 4-50
- nocto, 4-50
- nointr, 4-50
- nosuid, 4-51
- port, 4-51
- retrans, 4-51
- retry, 4-51
- ro, 4-51
- rsize, 4-51
- rw, 4-51
- soft, 4-51
- suid, 4-52
- timeo, 4-52

mountd server, defined, 4-10

Mounts

- Automatic, 4-53
- Clusters, C-2
- Guidelines, 4-48
- Hard, defined, 4-47
- Hard, via /etc/checklist, 4-53
- Hard, via mount command, 4-56
- Manual, 4-56
- see* Mount defaults
- Soft, defined, 4-47
- Soft, via /etc/checklist, 4-53
- Soft, via mount command, 4-56

N

Named pipes

- Defined, 2-6
- mknod, created with, 2-7
- NFS Services vs. Local HP-UX, A-2

Netgroups

- NFS configuration, 4-32
- NIS configuration, 7-9

Network Information Service

- Configuration, 7-16
- Databases, 7-6
- Disabling of, 7-34
- Escape sequences, 7-8

- Log files, 7-43
- Maps, 7-7
- Troubleshooting, 11-9
- see also*: Troubleshooting NIS
- Verification of, 7-33
- Network Information Service (NIS)
 - Advantages, 2-14
 - Clients, 2-17, 10-17
 - Commands, 10-18
 - Concepts, 2-16
 - Defined, 2-14, 10-16
 - Disadvantages, 2-15
 - Domains, 2-18, 10-17
 - Maps, 2-17, 10-16
 - Master server, 2-18, 10-17
 - Servers, 2-17, 10-17
 - Slave server, 2-18, 10-17
 - Structure, 2-16
- Network Lock Manager
 - Defined, 2-13
 - fcntl(), mapping user calls, 6-1
 - Introduction, 6-1
 - lockf(), system call interface, 6-1
 - Locking protocol, 6-4
 - Network locking service, starting, 6-3
 - Network locking service, structure of, 6-2
 - Network Status Monitor, 6-1, 6-5
 - Network Status Monitor, defined, 2-13
 - rpc.lockd, 2-13
 - rpc.lockd, network lock manager, 6-1
 - rpc.statd, 2-13
 - rpc.statd, network status monitor, 6-1
- Network memory, 4-5
- Network Status Monitor, 2-13, 6-5
 - see* Network Lock Manager
- NFS Automounter, 2-25
 - Advantages, 2-25
 - Disadvantages, 2-25
- NFS Clients, defined, 2-1
- NFS Configuration, 4-15
 - Becoming an NFS Client, using manual method, 4-44
 - Becoming an NFS server, using manual method, 4-20
 - Clusters, C-2
 - Configuration files, 4-6
 - Defined, 4-11

- Edit /etc/checklist, manually, 4-53
- Edit /etc/exports, manually, 4-34
- Edit /etc/hosts, manually, 4-27
- Edit /etc/inetd.conf, manually, 4-23
- Edit /etc/netfsrc, manually, 4-44
- Edit /etc/netgroup, manually, 4-32
- Edit /etc/netnfsrc, manually, 4-20
- Guidelines, 4-5
- HP-UX cluster environment, C-1
- Memory, 4-5
- see also:* NFS Maintenance
- Overview, 4-1
- Overview of SAM, 4-14
- Rebooting, using manual method, 4-44, 4-58
- Security, 4-23
- Security, manually edit /usr/adm/inetd.sec, 4-25
- Security, RPC services, 4-27
- Servers, setting number of remote connections, 4-25
- Servers, specifying access to services, 4-25
- Set UIDs and GIDs, 4-12
- Tips for using SAM, 4-14
- Troubleshooting, 11-9
- NFS export options, 4-39
- NFS Maintenance
 - Clock skews, 4-66
 - Clusters, C-2
 - Directories or Files, preventing access, 4-63
 - NFS servers, maintaining, 4-69
 - NFS servers, planning downtime, 4-69
 - NFS servers, reacting to unplanned downtime, 4-70
 - Overview, 4-59
 - Prevent NFS file access, 4-61
 - Server directories or files, preventing access, 4-63
 - Update software, using /etc/update, 4-64
- NFS servers
 - List of, 4-10
 - Security, 4-23, 4-25
- NFS Servers, defined, 2-1
- NFS Services
 - see also:* Commands
 - Common commands, 10-5
 - Components of, 2-2
 - HP NFS Services vs. Local HP-UX, A-1
 - Moving from RFA to NFS, B-1
 - see also:* NFS Services vs. Local HP-UX

- Overview, 2-1
- Remote file access, 10-6, B-1
- RFA to NFS, changing scripts, B-3
- NFS Services vs. Local HP-UX
 - append mode, A-1
 - chacl, A-1
 - Device files, A-1
 - File locking, A-1
 - getacl system call, A-1
 - Group membership, A-2
 - lseek, A-2
 - mknod command, A-2
 - Mount points, A-2
 - Named pipes, A-2
 - pathconf/fpathconf, A-6
 - Reading directories, A-2
 - setacl system call, A-3
 - setaclentry library routine, A-3
 - Superuser permission, A-3
 - System time, A-3
 - Unlinking, A-4
 - yppasswd vs. passwd, A-5
- nfsd daemon
 - Defined, 4-8
- NIS clients, 2-17, 10-17
 - Alteration of, 7-22
 - Automatic start, 7-26
 - Configuration, 7-21
 - Manual start, 7-26
 - Troubleshooting, 11-46
- NIS Configuration, 7-16
 - Clients, 7-21
 - Clusters, C-3
 - Master servers, 7-17
 - Propagate maps, 7-30
 - Slave server, 7-27
- NIS domains, 2-18, 10-17
- NIS Maintenance, 7-34

- Adding new users, 7-39
- Adding servers, 7-38
- Log files, 7-43
- Master servers, changing, 7-40
- Modifying maps, manually, 7-36
- Modifying NIS maps, 7-35

- Non-standard maps, 7-44
- Password, 7-41
- NIS password, 7-41, 10-21–10-22, D-1
- NIS servers, 2-17, 10-17
- Adding of, 7-38
- Maintaining, 7-38

O

on command

- Debug mode, -d option, 10-15
- Defined, 10-14
- Interactive mode, -i option, 10-15
- No input mode, -n option, 10-15
- Syntax, 10-14

P

Password security, D-1

PC-NFS servers

- Creating using manual method, 4-44
- Creating, using manual method, 4-20

pcnfsd daemon, defined, 4-8

portmap daemon, defined, 4-9

Propagate maps, 7-30

R

Reboot system

- Using manual method, 4-44, 4-58

Remote Execution Facility

see REX

Remote file access

- Defined, for NFS, 2-4
- Device files, defined, 2-7
- Limiting, 4-32, 4-34
- Migration of NS to NFS, B-1
- mknod, creating named pipes, 2-7
- Named pipes, defined, 2-6
- Preventing, 4-61
- Procedure, for NFS, 10-6

Remote Procedure Call

see RPC

Remote services

- Setting access to, 4-25

- Setting maximum number of, 4-25
- REX (Remote Execution Facility)
 - \$HOME/.rhosts, adding stricter security to rexd, 5-8
 - /etc/hosts.equiv, adding stricter security to rexd, 5-7
 - /etc/inetd.conf, configuring rexd, 5-6
 - /usr/adm/inetd.sec, reducing system security, 5-9
 - /usr/etc/rpc.rexd, file containing rexd, 5-6
 - /usr/spool/rexd, in environment simulation, 5-5
 - Configuring rexd server, 5-6
 - Configuring rexd, -l option, 5-6
 - Configuring rexd, -m option, 5-7
 - Configuring rexd, -r option, 5-7
 - Defined, 2-9
 - Diagnostics, 5-10
 - Environment simulation, 5-5
 - Error messages, on command, 5-10
 - Error messages, rexd daemon, 5-11
 - Invoking debug mode, 5-4
 - Invoking interactive mode, 5-3
 - Invoking no-input mode, 5-3
 - Logging errors, 5-6
 - on command, 5-2
 - on command, configuration requirements, 5-4
 - on command, using -d option, 5-4
 - on command, using -i option, 5-3
 - on command, using -n option, 5-3
 - Overview, 5-1
 - Security limitations, 5-9
 - Security, adding stricter, 5-7
 - Specifying mount point directory, 5-7
 - Troubleshooting, 11-10, 11-60
 - see also:* Troubleshooting REX
- rexd server, configuring, 5-6
- RPC (Remote Procedure Call)
 - Compiler, RPCGEN, 2-11
 - Defined, 2-10
 - Network Lock Manager, mapping user calls, 6-1
- RPC services
 - /etc/inetd.conf entries, 4-24
 - /usr/adm/inetd.sec entries, 4-27
 - Activation of, 4-23
 - Security, 4-23, 4-27
- RPCGEN, RPC compiler, 2-11
- rpcinfo command, 10-8
- rstatd server, defined, 4-10

- rup command, 10-10
- rusers command, 10-11
- rusersd server, defined, 4-10
- rwalld server, defined, 4-10

S

SAM (System Administration Manager)

- Overview for creating NFS servers and clients, 4-14
- Tips for using, 4-14

Security

- Edit /usr/adm/inetd.sec, manually, 4-25
- NIS, master servers, 7-18
- Password, D-1
- REX restrictions, 5-9
- RPC services, 4-23, 4-27

Servers

- Creating NFS servers, using manual method, 4-20
- Creating NFS servers, using SAM, 4-14-4-15
- Defined, for NFS, 2-1
- mountd, defined, 4-10
- NFS, 4-10
- NFS, maintaining, 4-69
- NFS, reacting to unplanned downtime, 4-70
- NIS, 2-17, 10-17
- NIS masters, 2-18, 10-17
- NIS slaves, 2-18, 10-17
- PC-NFS, becoming a server using manual method, 4-20, 4-44
- Preventing access, 4-63
- rstatd, defined, 4-10
- rusersd, defined, 4-10
- rwalld, defined, 4-10
- sprayd, defined, 4-10
- Stateless, defined, 2-1
- setuid, mount defaults, 4-52
- showmount command, 10-13
- Slave servers
 - Automatic start, 7-29
 - Manual start, 7-29
 - NIS, configuration, 7-27
- Soft mounts
 - Defined, 4-47
 - soft, option, 4-51
 - via /etc/checklist, 4-53
 - via mount command, 4-56

Software

- Install using /etc/update, 4-64
- Software Installation, 3-6
- sprayd server, defined, 4-10
- Stateless servers, 2-1
- System Administration Manager
 - see SAM

T

Troubleshooting

- Clusters, C-3
 - Configuration, 11-8
 - Error messages, 11-12
 - Flowchart formats, 11-13
 - Hardware, 11-8
 - Initial steps, 11-7, 11-60
 - Key terms, 11-2
 - Named pipe problems, 11-85
 - Network communication, 11-9
 - Network problems, 11-7
 - NFS configuration, 11-9
 - NIS Clients, 11-46
 - NIS configuration, 11-9
 - References, 11-5
 - REX configuration, 11-10
 - Stale File Error Messages, 11-12
 - see also: Troubleshooting NFS
 - see also: Troubleshooting NIS
 - see also: Troubleshooting REX
 - see also: Troubleshooting VHE
 - Unsolved problems, 11-12
 - VHE, initial steps, 11-48
 - ybind, 11-44
- ### Troubleshooting NFS
- Access is restricted, 11-28
 - Initial steps, 11-15
 - Mount failure, 11-18
 - Performance problems, 11-34
 - Programs hang, 11-31
 - Server not responding, 11-21, 11-24
- ### Troubleshooting NIS
- Client problems, 11-46
 - Incorrect maps, 11-39
 - Initial steps, 11-37

ypbind problems, 11-44

ypserv problems, 11-42

Troubleshooting REX

Command, not found, 11-79

Device file problems, 11-85

Initial steps, 11-60, 11-62

Mount point, 11-77

Permission denied, 11-81

Server, access denied, 11-74

Server, connection problems, 11-66

Server, mount daemon problems, 11-72

Text file busy, 11-83

Unknown host, 11-64

User ID, access denied, 11-70

User ID, not valid, 11-68

Troubleshooting VHE

/etc/passwd file, accuracy, 11-52

/etc/passwd file, consistency, 11-54

/etc/vhe_list file, accuracy, 11-52

/etc/vhe_list file, consistency, 11-54

Home node, 11-50

Initial steps, 11-48

vhe_mounter, error messages, 11-58

vhe_mounter, execution, 11-56

U

UID, Setting of, 4-12

umount command

Preventing access to server directories or files, 4-64

Unmounting directory from client, 4-61

Unmounting directories or files, manually, 4-61

update command, installing software, 3-6

update program, installing software, 4-64

/usr/adm/inetd.sec file

Defined, 4-7

REX, reducing security, 5-9

/usr/adm/inetd.sec file

Manually edit, 4-25

/usr/etc/rpc.rexd, file containing rexd, 5-6

/usr/spool/rexd, in REX environment simulation, 5-5

V

Variables

- `_PC_CHOWN_RESTRICTED`, A-6
- `_PC_LINK_MAX`, A-6
- `_PC_MAX_CANON`, A-6
- `_PC_MAX_INPUT`, A-6
- `_PC_NAME_MAX`, A-6
- `_PC_NO_TRUNC`, A-6
- `_PC_PATH_MAX`, A-6
- `_PC_PIPE_BUF`, A-6
- `_PC_VDISABLE`, A-6

VHE

see Virtual Home Environment

VHE advanced usage

- Alternate mount points, 8-16
- `altlogin`, `login`, 8-14
- `mounter`, `login`, 8-14
- Using for mail, 8-16

VHE Configuration

- `/etc/passwd` file, 8-3
- `/etc/vhe_list` file, 8-3
- Allowing for background NFS mounts, 8-11
- Compare files, `/etc/newconfig` vs existing, 8-4
- Create `/etc/vhe_list` file, 8-5
- Determine directories, 8-4
- Distribute `/etc/vhe_list` and `/etc/passwd` files, 8-9
- Execute `/usr/etc/vhe/vhe_mounter` script, 8-9
- Interactions with NIS, 8-3
- Overview, 8-2
- Preparation steps, 8-3
- Refinements, 8-11
- Update `/etc/exports` file, 8-8
- Update `/etc/passwd` file, 8-7
- Verify configuration, 8-10

see also: Virtual Home Environment (VHE)

VHE Maintenance

- Adding or deleting nodes, 8-13
- Unmounting directories or files, 8-12

Virtual Home Environment (VHE)

- Advanced usage, 8-14
- Advantages, 2-20
- Concepts, 2-23–2-24
- Create `/etc/vhe_list` file, 8-5
- Defined, 2-20

- Determine directories, 8-4
- Disadvantages, 2-22
- Distribute /etc/vhe_list and /etc/passwd files, 8-9
- Execute /usr/etc/vhe/vhe_mounter script, 8-9
- Maintenance, 8-12
- Overview, 8-1
- Preparing for configuration, 8-3
- Troubleshooting, 11-48
 - see also:* Troubleshooting VHE
- Update /etc/exports file, 8-8
- Update /etc/passwd file, 8-7
- Verify configuration, 8-10
 - see also:* VHE advanced usage
 - see also:* VHE Maintenance

W

- wsize, mount defaults, 4-52

X

- XDR (External Data Representation), 2-12

Y

- Yellow Pages (YP)

- see* Network Information Service

- ypbind command, 7-14
- ypcat command, 7-14, 10-19
- ypinit command, 7-14
- ypmatch command, 7-14, 10-20
- yppasswd command, 7-15, 7-41, 10-21–10-22
- yppasswd installation guidelines, 7-41
- yppasswdd command, 7-15
- yppoll command, 7-15
- yppush command, 7-31
- ypservc command, 7-15
- ypwhich command, 10-23
- ypxfr command, 7-15, 7-30, 7-32

Reader Comment Sheet

HP 9000 Computers

Installing and Administering NFS Services

B1013-90009, E0992

We welcome your evaluation of this manual. Your comments and suggestions help us to improve our publications. Please explain your answers under "Comments", below. Use additional pages if necessary.

Please answer the following questions:

Is this manual well organized?	Yes	No
Is the information technically accurate?	Yes	No
Are the concepts and wording easy to understand?	Yes	No
Is the format of the manual convenient in size, arrangement, and readability?	Yes	No

Comments:

Date _____

Name _____

Company _____

Address _____

City & State _____

Zip Code _____



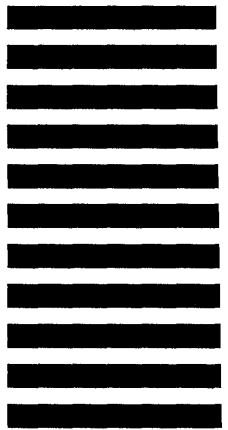
NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATE

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1070 CUPERTINO, CA

POSTAGE WILL BE PAID BY ADDRESSEE

Hewlett-Packard Company
Information Networks Division
19420 Homestead Road
Cupertino, CA 95014
Attn: Technical Marketing Dept.



Fold Here

Tape

Please do not staple

Tape

**Customer Order No.
B1013-90009**

Copyright © 1992
Hewlett-Packard Company
Printed in USA 09/92

Manufacturing No.
B1013-60009
Mfg. number is for HP internal use only



B1013-60009