Tru64 UNIX

Network Administration: Services

Part Number: AA-RPPCB-TE

September 2002

Product Version: Tru64 UNIX Version 5.1B or higher

This manual is intended for experienced system or network administrators. It describes the tasks for configuring your system to operate in a network, for configuring the network services, and for day-to-day management of the network, network interfaces, and network services. This manual also includes information for solving problems that might arise while using the network and network services.

Hewlett-Packard Company
Palo Alto, California

Contents

3 Network Information Service

4 Network File System

5 UNIX-to-UNIX Copy Program

E sendmail Error Messages

F Host Resources MIB Implementation

G Format of DNS Data File Entries

Index

Examples

Figures

Tables

About This Manual

This manual describes how to configure and manage network applications and services, and solve network problems that might arise on systems running the Tru64 UNIX operating system software.

This manual assumes that the operating system software and the appropriate networking subsets are installed.

Audience

This manual is intended for system and network adminstrators responsible for configuring and managing network services. Administrators are expected to have knowledge of operating system concepts, commands, and configuration. It is also helpful to have knowledge of Transmission Control Protocol/Internet Protocol (TCP/IP) networking concepts and network configuration; this manual is not a TCP/IP networking tutorial.

New and Changed Features

The *Network Administration: Services* manual contains new and revised sections, including:

- An updated Network Information Service (NIS) chapter, with revised sections on managing an NIS server (*Section 3.4*)
- An updated Network File System (NFS) chapter, with new and revised sections on configuring the AutoFS service (*Section 4.6.3*)
- An updated appendix on writing Automount and AutoFS maps, with a new section describing the differences between Automount and AutoFS behavior (*Section A.4*)
- An updated problem solving chapter that contains a new section on AutoFS (*Section 9.9*)
- An updated appendix that contains scripts for managing NIS slave servers (*Appendix B*)
- An updated appendix on DNS data file entries (*Appendix G*)

Organization

The *Network Administration: Services* manual is divided into several chapters, each of which contains information about configuring a different

service or application. The manual also includes appendixes that contain supplemental information.

The following list describes the content in more detail:

Chapter 1	Describes network administration and the components that this manual covers.
Chapter 2	Describes the tasks to administer the Domain Name System (DNS)
Chapter 3	Describes the tasks to administer the Network Information Service (NIS)
Chapter 4	Describes the tasks to administer the Network File System (NFS)
Chapter 5	Describes the tasks to administer the UNIX-to-UNIX Copy Program (UUCP)
Chapter 6	Describes the tasks to administer the Network Time Protocol (NTP)
Chapter 7	Describes the tasks to administer the mail environment
Chapter 8	Describes the Simple Network Management Protocol (SNMP)
Chapter 9	Describes how to diagnose network and network service problems
Chapter 10	Describes the various diagnostic tools available to help solve problems
Chapter 11	Describes how to test DNS servers and resolve DNS server problems
Chapter 12	Describes how to report problems to HP and the information you need to provide
Appendix A	Describes how to write Automount and AutoFS maps
Appendix B	Provides two scripts you can copy for adding NIS slave servers to and removing NIS slave servers from an NIS domain
Appendix C	Describes NFS error messages and provides possible explanations
Appendix D	Describes `uucp` error messages and provides possible explanations
Appendix E	Describes `sendmail` error messages and provides possible explanations
Appendix F	Describes the Tru64 UNIX host MIB implementation, including sample data
Appendix G	Describes the format of DNS data file entries

Related Documents

For more information about Tru64 UNIX networking and communications, see the following books:

- *Network Administration: Connections*

  Provides information about the network connections over which the networking services and applications covered in this manual run. Explains how to configure and manage the following connections and transports:

  – Basic network connections, including Ethernet, Token Ring, and Fiber Distributed Data Interface (FDDI) interfaces, automatic network adapter failover (NetRAIN), and network daemons

  – Internet Protocol Version 6 (IPv6) and Mobile IPv6

  – Internet Protocol Security (IPsec)

  – Asynchronous Transfer Mode (ATM)

  – Dynamic Host Configuration Protocol (DHCP)

  – Point-to-point connections, including Serial Line Internet Protocol (SLIP) and Point-to-Point Protocol (PPP)

  – Local Area Transport (LAT)

- *BIND Configuration File Guide*

  Provides information about how to manually create and edit the `named.conf` configuration file on systems that use the DNS/BIND for address resolution. This document is available in HTML format on the Tru64 UNIX Documentation CD-ROM.

- *Command and Shell User's Guide*

  Introduces users to the basic uses of commands and shells in the operating system.

- *Sendmail Installation and Operation Guide*

  Provides additional information about using the `sendmail` command. This document is available in PDF format on the Tru64 UNIX Documentation CD-ROM.

- The *sendmail* guide by O'Reilly & Associates

  Provides additional information about using the `sendmail` command.

- Request for Comments (RFC)

  Many sections of this book refer to RFCs (for example, RFC 1577) for more information about certain networking topics. These documents publicize Internet Standards, new research concepts, and status memos about the internet. You can access the full range of RFC documents and

more information about the Internet Engineering Task Force (IETF) at the following URL:

http://www.ietf.org

- Best Practices

  Tru64 UNIX Best Practices describe some networking concepts and tasks, as well as other topics. You can find these documents on the Tru64 UNIX Publications Home Page at the following URL:

  http://www.tru64unix.compaq.com/docs/

Icons on Tru64 UNIX Printed Manuals

The printed version of the Tru64 UNIX documentation uses letter icons on the spines of the manuals to help specific audiences quickly find the manuals that meet their needs. (You can order the printed documentation from HP.) The following list describes this convention:

G	Manuals for general users
S	Manuals for system and network administrators
P	Manuals for programmers
R	Manuals for reference page users

Some manuals in the documentation help meet the needs of several audiences. For example, the information in some system manuals is also used by programmers. Keep this in mind when searching for information on specific topics.

The *Documentation Overview* provides information on all of the manuals in the Tru64 UNIX documentation set.

Reader's Comments

HP welcomes any comments and suggestions you have on this and other Tru64 UNIX manuals.

You can send your comments in the following ways:

- Fax: 603-884-0120 Attn: UBPG Publications, ZKO3-3/Y32
- Internet electronic mail: `readers_comment@zk3.dec.com`

  A Reader's Comment form is located on your system in the following location:

  `/usr/doc/readers_comment.txt`

Please include the following information along with your comments:

- The full title of the manual and the order number. (The order number appears on the title page of printed and PDF versions of a manual.)
- The section numbers and page numbers of the information on which you are commenting.
- The version of Tru64 UNIX that you are using.
- If known, the type of processor that is running the Tru64 UNIX software.

The Tru64 UNIX Publications group cannot respond to system problems or technical support inquiries. Please address technical questions to your local system vendor or to the appropriate HP technical support office. Information provided with the software media explains how to send problem reports to HP.

Conventions

This document uses the following typographic conventions:

%	A percent sign represents the C shell system prompt.
$	A dollar sign represents the system prompt for the Bourne, Korn, and POSIX shells.
#	A number sign represents the superuser prompt.
% **cat**	Boldface type in interactive examples indicates typed user input.
file	Italic (slanted) type indicates variable values, placeholders, and function argument names.
[|]	
{ | }	In syntax definitions, brackets indicate items that are optional and braces indicate items that are required. Vertical bars separating items inside brackets or braces indicate that you choose one item from among those listed.
. . .	In syntax definitions, a horizontal ellipsis indicates that the preceding item can be repeated one or more times.
cat(1)	A cross-reference to a reference page includes the appropriate section number in parentheses.

	For example, `cat`(1) indicates that you can find information on the `cat` command in Section 1 of the reference pages.
Return	In an example, a key name enclosed in a box indicates that you press that key.
Ctrl/*x*	This symbol indicates that you hold down the first named key while pressing the key or mouse button that follows the slash. In examples, this key combination is enclosed in a box (for example, Ctrl/C).

1

Overview to Network Administration

Network administration comprises those tasks that deal with setting up and configuring network interfaces, software, and daemons, and those tasks that deal with the day-to-day management of those interfaces, software, and daemons, including solving problems that might arise.

This chapter describes:

- How to use this manual in the day-to-day management of your network (Section 1.1)
- Several utilities and methods you can use to administer network components (Section 1.2)

1.1 Introduction to *Network Administration: Services*

This manual describes the administration of the following:

- Domain Name System (DNS) (Chapter 2)
- Network Information Service (NIS), formerly named Yellow Pages (Chapter 3)
- Network File System (NFS) (Chapter 4)
- UNIX-to-UNIX Copy Program (UUCP) (Chapter 5)
- Network Time Protocol (NTP) (Chapter 6)
- Mail system, including sendmail, Post Office Protocol (POP) and Internet Message Access Protocol (IMAP) (Chapter 7)
- Simple Network Management Protocol (SNMP) (Chapter 8)

Information regarding network connections and transports is maintained in a separate volume, *Network Administration: Connections*.

Day-to-day management varies with each network service, as each service provides different capabilities. Typically, management involves making small changes and adjustments, such as adding NIS user accounts, mounting remote file systems or directories, obtaining status information, and setting up automatic maintenance scripts. Chapters 2–8 of this manual describe specific tasks, presenting the generic steps required to perform the tasks followed by examples and additional information.

In addition to the day-to-day management of network services and applications, this manual contains information to help you solve problems that might occur. Problem solving is handled differently from administration because it is not something that you have to do every day.

Unlike the administration chapters, problem-solving chapters are structured according to specific problems. Within each problem section are the steps to resolve the problem.

The key to successful problem solving is in isolating the source of the problem. Frequently, complex networks and interactions between network services make this difficult to do. If you encounter a problem, whether by error message or event (for example, slow response), do the following:

1. Check your system, its network interface, and connections to the network.
2. Check the network and your system's ability to reach a remote system.

Most problems can be solved after you perform these two steps. If not, go to the appropriate problem-solving section and follow the steps.

1.2 Administrative Methods

The following sections provide a brief overview of the methods for administering networking components in the operating system. As explained in Section 1.2.4, it is best to not to edit configuration files manually for network configuration tasks. Instead, it is highly recommended that you use the SysMan Menu utility whenever possible.

1.2.1 SysMan Menu

The SysMan Menu utility enables you to administer your system locally via a graphical user interface or command-line interface, or remotely via the World Wide Web. It provides a single, hierarchical menu interface that allows you to quickly find and invoke suitlets (integrated utilities) to perform the most common management tasks.

In this manual, wherever the SysMan Menu utility is mentioned in relation to configuration tasks, it is presumed that you know how to invoke it. To invoke the SysMan Menu utility from CDE, do the following:

1. Select the Application Manager icon on the CDE front panel.
2. Select the System_Admin application group icon.
3. Select the SysMan Menu. The SysMan Menu is displayed and lists various system management tasks.

If you are not using CDE, you can invoke the SysMan Menu in one of the following ways:

```
# /usr/bin/sysman
```

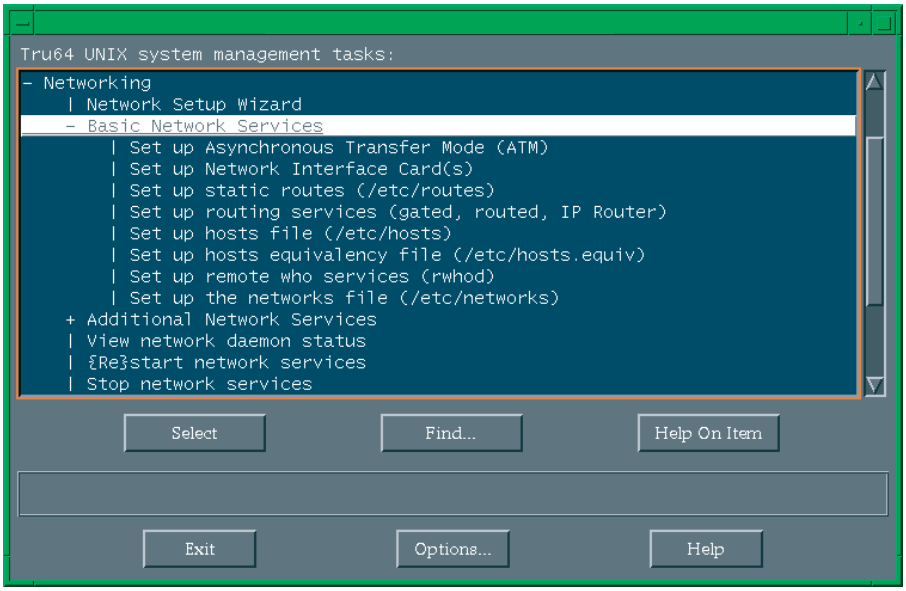
From a character-cell terminal or terminal window, for curses mode, enter:

```
# sysman -ui cui
```

After you invoke the SysMan Menu, double-click on menu items to select them. Or, on a system without graphics capabilities, use the arrow keys and the Enter key to select items. Many menu items will expand to offer more choices. Navigate the menu until you find the desired suitlet.

In Figure 1–1, the user selects the Basic Network Services menu item, which expands to reveal the suitlets for configuring network adapters and other basic networking components.

Figure 1–1: SysMan Menu



To exit the SysMan Menu, select Exit. On a system without graphics capabilities, use the Tab key to move the cursor to Exit, then press the Enter key.

For more information about the SysMan Menu, see *System Administration*, `sysman`(8), and the online help.

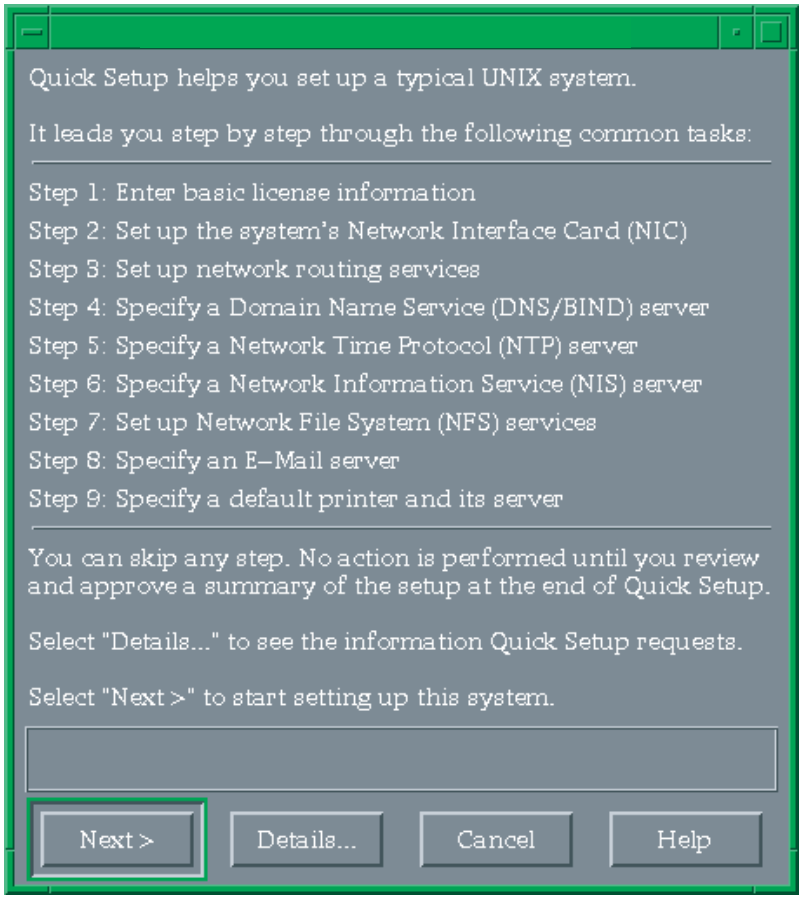
1.2.1.1 Quick Setup

The SysMan Menu includes a Quick Setup utility that you can use to configure basic components and services on a client system. The Quick Setup utility starts automatically when the system boots following a full installation of the operating system. However, to use the utility at any time, invoke the SysMan Menu and select General Tasks→Quick Setup, or enter the following command on a command line:

```
# /usr/bin/sysman quicksetup
```

The Quick Setup utility, as shown in Figure 1–2, is displayed.

Figure 1–2: Quick Setup



The utility leads you through the displayed configuration steps, many of which prepare your system for operation on a network. Enter the information for each step of the process and select Next to display the subsequent step. You can move back and forth through the steps if you

have missed something. No information is saved until you confirm the configuration by selecting Finish in the last step.

If necessary, you can configure additional components or modify your configuration after you use the utility. For more information about the Quick Setup utility, see the online help.

1.2.1.2 Network Setup Wizard

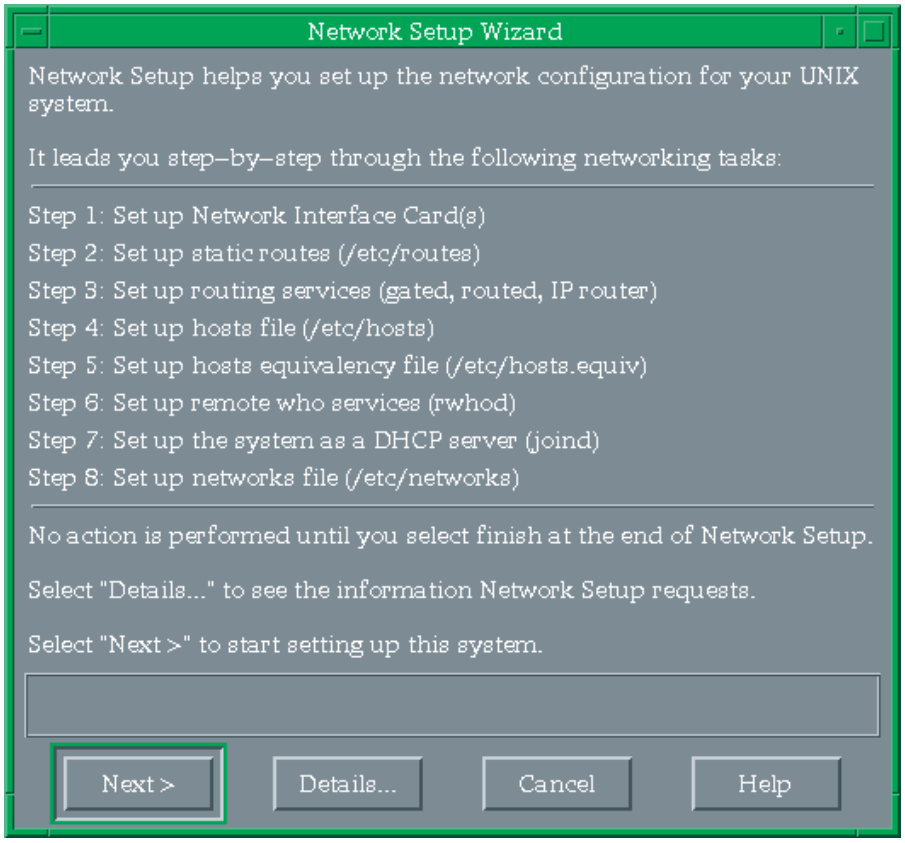
The SysMan Menu also includes a Network Setup Wizard utility that you can use to configure network components on your system. You can invoke the configuration suitlets through the SysMan Menu to configure basic network services on an individual basis, or you can use the Network Setup Wizard, which leads you step-by-step through the setup process for all of the basic network services.

To use the Network Setup Wizard, invoke the SysMan Menu and select Networking→Network Setup Wizard, or enter the following command on a command line:

```
# /usr/bin/sysman net_wizard
```

The Network Setup Wizard utility, as shown in Figure 1–3, is displayed.

Figure 1–3: Network Setup Wizard



The utility leads you through the displayed configuration steps. Enter the information for each step of the process and select Next to display the subsequent step. You can move back and forth through the steps if you have missed something. No information is saved until you confirm the configuration by selecting Finish in the last step.

If necessary, you can configure additional components or modify your configuration after you use the utility. For more information about the Network Setup Wizard utility, see the online help.

1.2.1.3 Command-Line Integration

The SysMan Menu allows you to access and manipulate many configuration options directly from the command line. This feature is particularly useful for administrators who want to create site-specific shell scripts to perform configuration tasks.

To use the command-line interface, invoke the `sysman -cli` command. For the command's arguments, specify the component and group on which you want to operate, and the action you want to perform.

For example, suppose you want to list all of the entries in the `/etc/hosts` file. You would enter the following command:

```
# sysman -cli -list values -comp networkedSystems \
 -group hostMappings
```

You could also add a host to the file by entering this command:

```
# sysman -cli -add row -comp networkedSystems \
-group hostMappings -data "{queen} \
{DNS server} {18.240.32.40} {queen.abc.xyz.com}"
```

You can change an existing value in the file, like an IP address, as follows:

```
# sysman -cli -set val -comp networkedSystems \
-group hostMappings -attr networkAddress="18.240.32.45" \
-key1 queen.abc.xyz.com -key2 18.240.32.40
```

For more information about this command line interface for the SysMan Menu, see *System Administration* and `sysman_cli`(8).

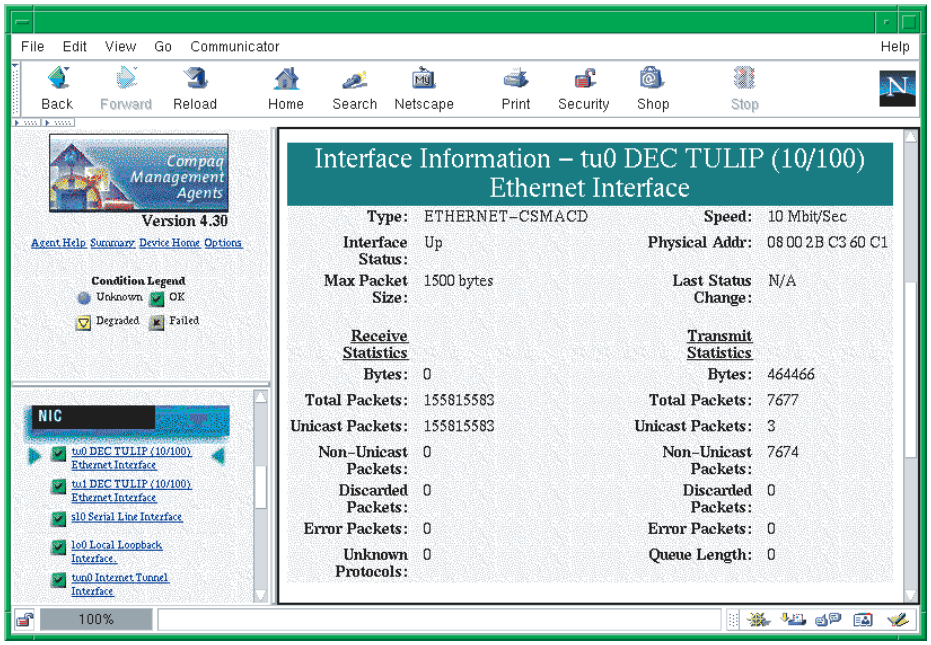
1.2.2 Compaq Insight Manager

Compaq Insight Manager is a Web-based system management utility. It consists of two different components: the Management Agents, which run on many different operating systems (including Tru64 UNIX), and the Management Console, which runs exclusively on Microsoft Windows NT.

By enabling the Compaq Management Agents on your Tru64 UNIX systems, you can provide a conduit for communication between these systems and the World Wide Web. Once enabled, this conduit allows you to access information about the configuration of your systems and their peripherals from a Web browser on any system. In some Java-enabled Web browsers, you can also invoke the SysMan Menu through this interface to manage these systems.

Figure 1–4 shows an example of using the Management Agents to obtain statistics for an Ethernet network adapter.

Figure 1–4: Compaq Management Agents



Using the Compaq Insight Manager XE Management Console, you can view and manage your systems as well as many standalone devices (such as printers, routers, and more) on your network. The Management Console is especially useful for managing heterogeneous environments, as it can communicate with the Management Agents for all of the supported operating systems and environments.

For more information about Compaq Insight Manager, see `insight_manager`(5) and *System Administration*.

1.2.3 Other Interfaces

The operating system includes alternative system administration applications, some that require graphics capabilities and others that allow you to configure your system from the command line. This manual mentions these optional utilities, when available, in relation to specific configuration tasks.

See *System Administration* for a comprehensive list of the utilities that are available. See the reference pages and online help for more information about each utility.

1.2.4 Manually Editing Configuration Files

Some sections of this manual describe the system files that are updated or modified when you perform an administrative task. Experienced UNIX administrators might prefer to administer their systems by manually editing these files, as opposed to invoking the documented utility; however, it is strongly recommended that you use the appropriate utilities to update the system files so that the structure of these files is preserved.

Important considerations are:

- Context-Dependent Symbolic Links (CDSLs)

  Many system files now exist as special symbolic links (CDSLs) created to facilitate TruCluster Server clusters. The links are transparent to most users, but if the links are broken, the system cannot join a cluster in the future without re-creating them. This manual mentions a few of the CDSLs, especially when you must create them manually. See the `hier`(5) reference page for a complete list of the CDSLs in the file system. See *System Administration* for more information.

- Binary databases, configuration definitions

  Many system components write data to both text and binary files, and their administrative utilities often re-create the binary files. Other system information is often preserved so that when you update your system, it can be recovered and reused, saving you time and effort.

- Latent support for clusters

  Individual systems are capable of joining TruCluster Server clusters, and many system files have been modified to provide latent support for clusters. For example, the `rc.config` file now has two related files, `rc.config.common` and `rc.config.site`, which can store run-time configuration variables. Altering these files with the `rcmgr` utility ensures the integrity and consistency of these files.

- Update installation

  During the update installation process, changed information is merged into existing system files. The `.new..*` and `.proto..*` files might be important in this process. Refer to the *Installation Guide* for more information.

In many cases, the SysMan Menu utility is the best alternative to manually editing system files, thus it is the utility that is most frequently covered in this manual.

1.2.5 Installation and Configuration Cloning

The operating system includes two features, Installation Cloning and Configuration Cloning, that allow you to minimize the amount of manual

intervention that is necessary to install and configure systems. These features are particularly useful if you need to set up many identical systems in the same way, because they allow you to capture the configuration of a working system in configuration description files (CDFs) and use those files to install and configure subsequent systems.

See *Installation Guide — Advanced Topics* for more information.

2

Domain Name System

The Domain Name System (DNS) is a mechanism for resolving unknown host names and Internet Protocol (IP) addresses that originate from sites on your company's intranet or the Internet. A database lookup service that is part of the DNS daemon searches for the unknown hosts in local and remote `hosts` databases, which are distributed across the network by the DNS.

The implementation of DNS in Tru64 UNIX is based on Version 8.2.2 of the Berkeley Internet Name Domain (BIND) service, which is maintained by the Internet Software Consortium (ISC).

If you want to install BIND Version 9, you can obtain the software from the Internet Express for Tru64 UNIX kit (formerly Open Source Internet Solutions), a collection of popular Open Source software that HP distributes on a CD-ROM. For more information about Internet Express, see the following URL:

http://tru64unix.compaq.com/internet/prod_sol.htm

This chapter describes:

- The DNS environment (Section 2.1)
- A mechanism for remotely updating the DNS database from new DNS clients (Section 2.2)
- A mechanism for authenticating updates to the DNS database (Section 2.3)
- How to plan for your DNS configuration (Section 2.4)
- How to configure DNS servers and clients (Section 2.5)
- How to configure authentication on DNS servers (Section 2.6)
- How to deconfigure DNS servers and clients (Section 2.7)
- How to manage DNS servers and clients (Section 2.8)

For introductory information on DNS, see `bind_intro`(7). For additional information about BIND service, see Appendix G and the *BIND Configuration File Guide* (provided in HTML format on the Tru64 UNIX Documentation CD-ROM). You can also visit the Internet Software Consortium website at the following URL:

http://www.isc.org

Note

The BIND server daemon, `/usr/sbin/named`, supports AAAA (IPv6 address entry) lookups over IPv4 (AF_INET) connections only. The resolver and server have not been ported to IPv6, but IPv6 applications can make `getaddrinfo` and `getnameinfo` calls to retrieve the AAAA records. See the *Network Programmer's Guide* for information on using these routines.

For troubleshooting information, see Section 9.3 and Chapter 11 for servers and Section 9.4 for clients.

2.1 DNS Environment

In the DNS environment, systems can have the following roles:

- Master server — A system that is an authoritative source for information about a zone or zones and that maintains the master copy of the DNS database for the zone or zones.

  The master server runs the `named` daemon, answers requests from clients and other servers, caches information, and distributes the databases to slave servers.

- Slave server — A system that is an authoritative source for information about a zone or zones, but does not maintain the master copy of the DNS database for the zone or zones. Instead, a slave server loads its database files from the master server when the master server indicates that the files have been updated.

  A slave server runs the `named` daemon, provides backup for the master server, answers requests from clients and other servers, and caches information.

- Stub server — A master server that delegates authority for a specified subzone to a server local to the subzone.

  The stub server does not retain information in its configuration files about the machines in the specified subzone. Instead of searching the master DNS database, it queries the local server for information about machines in the subzone.

  Typically, stub service is implemented so that the administrator of a subzone can change the configuration of the subzone without affecting the configuration file on the master server.

- Caching-only server — A system that is not authoritative for any zones. This system runs the `named` daemon and responds to queries from other servers and clients by querying other servers for the information and

caching the information it receives. Information is stored until the data expires.

Typically, a caching-only server has direct access to the Internet and it answers queries exclusively about sites on the Internet.

- Forward-only server — A system that might be an authoritative source for information about a zone or zones, but is restricted as to how it obtains information about zones for which it is not authoritative.

  This system runs the `named` daemon and responds to queries from other servers and clients with information from its authoritative date and cache data. If the information is not present, the system forwards queries to a list of systems specified as forwarders in its `named.conf` file. The queries are forwarded to each forwarder system until the list is exhausted or the query is satisfied. Forward-only servers store the information they receive until the data expires.

  Typically, a forward-only server has restricted access to an intranet or the Internet. By providing a list of specific forwarders to contact, an administrator can prevent a forward-only server from attempting to contact servers that it cannot access.

- Client — A system that queries a server for host name and address information, interprets responses, and passes information to requesting applications. The client is also called a resolver. A client does not run the `named` daemon.

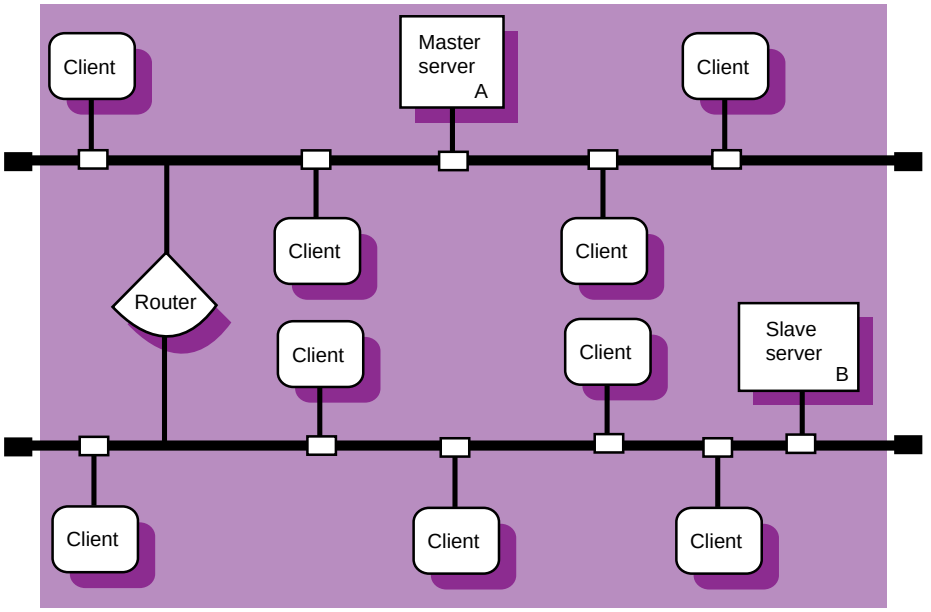
Note

Documentation for BIND prior to Version 8.1.1 referred to the master server as a primary server and the slave server as a secondary server. Though the terminology has changed, master and slave servers are still referred to as having primary and secondary authority, respectively, for zones.

DNS runs on each system in your network. You must decide what role each system will play in the DNS environment that you create. For each domain, select one host to be the master server; there can be only one master server for each domain. Select one or more hosts to be slave, stub, and caching-only servers. Configure the rest of the hosts as DNS clients.

Figure 2–1 shows a domain in which there are two servers, one on each subnet, and multiple clients. Server A, the master server, has primary authority for the zone and maintains the database files for the zone. Server B, the slave server, has secondary authority for the zone; it obtains a copy of the zone database from Server A and responds to queries from clients.

Figure 2–1: Sample Small DNS Configuration

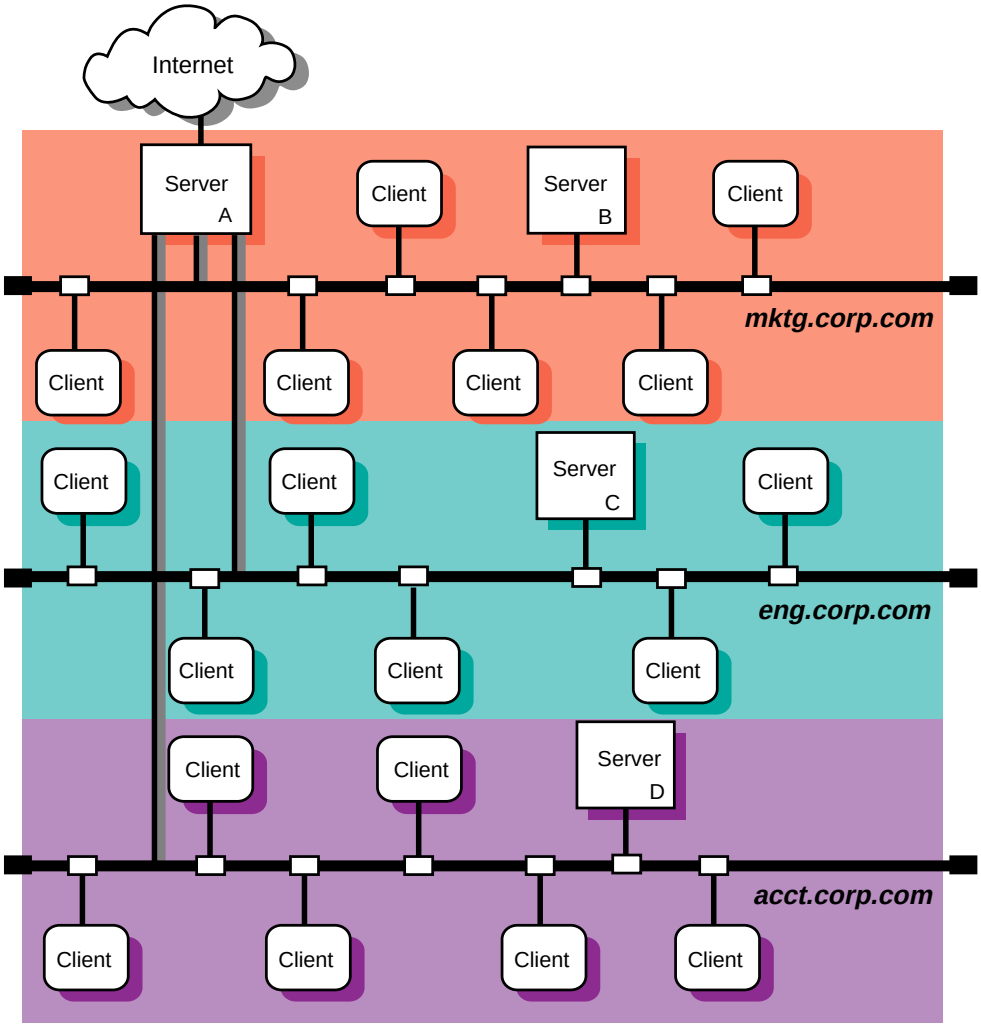


ZK-1162U-AI

Figure 2–2 shows a domain in which there are three zones: mktg.corp.com, eng.corp.com, and acct.corp.com. Server B is the master server for the mktg.corp.com zone and a slave server for the other two zones. It has primary authority for mktg.corp.com and secondary authority for each of the other two zones. Server C has primary authority for the eng.corp.com zone and secondary authority for each of the other two zones. Server D has primary authority for the acct.corp.com zone and secondary authority for each of the other two zones. Server A is both a router and a caching-only server. As a caching-only server, it caches information it receives from queries out of the parent domain.

In the same example, if the three zones were located in three different cities or countries, you could configure Server A at mktg.corp.com as a stub server for the other two remote zones. That way, all of the resource records for the remote sites would reside on servers (Server C and Server D) local to the eng.corp.com and acct.corp.com domains. The master server, Server A, would retain only the resource records for the name server that is local to each subdomain. Server A would query Server C and Server D for information about the machines in the eng.corp.com and acct.corp.com domains instead of searching its own master DNS database.

Figure 2–2: Sample Large DNS Configuration



ZK-1161U-AI

2.2 Dynamic Updates

Typically, whenever you connect a new host to a network, you need to rebuild the DNS database as explained in Section 2.8.2. If you do not update the DNS database, other computers on the network will not be able to resolve the new host's address.

However, some clients, particularly Tru64 UNIX systems that are configured for IPv6 networks and Microsoft Windows systems, can automatically update the DNS database for you. These clients support dynamic updates,

which allow hosts to inform the DNS master server that they are being added to or removed from the network. The clients specify their IP address and host name, and the `named` daemon automatically makes the appropriate changes in the DNS master data file. There is no need for intervention by the administrator of the master server, which saves the administrator a lot of effort in larger networks.

See Section 2.5.1.2 for information about configuring dynamic updates on the DNS master server.

2.3 Authentication of Dynamic Updates and Zone Transfers

DNS servers can provide cryptographic authentication of the data they receive from other systems. Authentication reduces the possibility that a rogue system can assume the identity of another system and send bogus DNS data file updates to servers.

The operating system provides support for symmetric cryptography, where two or more systems share a single private key for DNS authentication. A system that sends a DNS update to another system can use this key to generate a unique digital signature that corresponds to the data in the update. The system then attaches this signature to the update and sends the entire package to the target system. When the target system receives the signed update, it verifies the data by using the same private key to generate a second digital signature from the data. If the signatures match, the target system knows that the update came from the a trusted system and that it is safe to use.

You can use cryptographic authentication for many purposes, including:

- Secure dynamic updates — Allow the master server to authenticate the updates it receives from clients.
- Secure zone updates — Allow the master server to authenticate zone transfer requests it receives from the slave servers, and subsequently, allow slave servers to authenticate the zone transfers they receive from the master server.

For either of these applications, if the data is not correctly signed (by a trusted host), it is rejected.

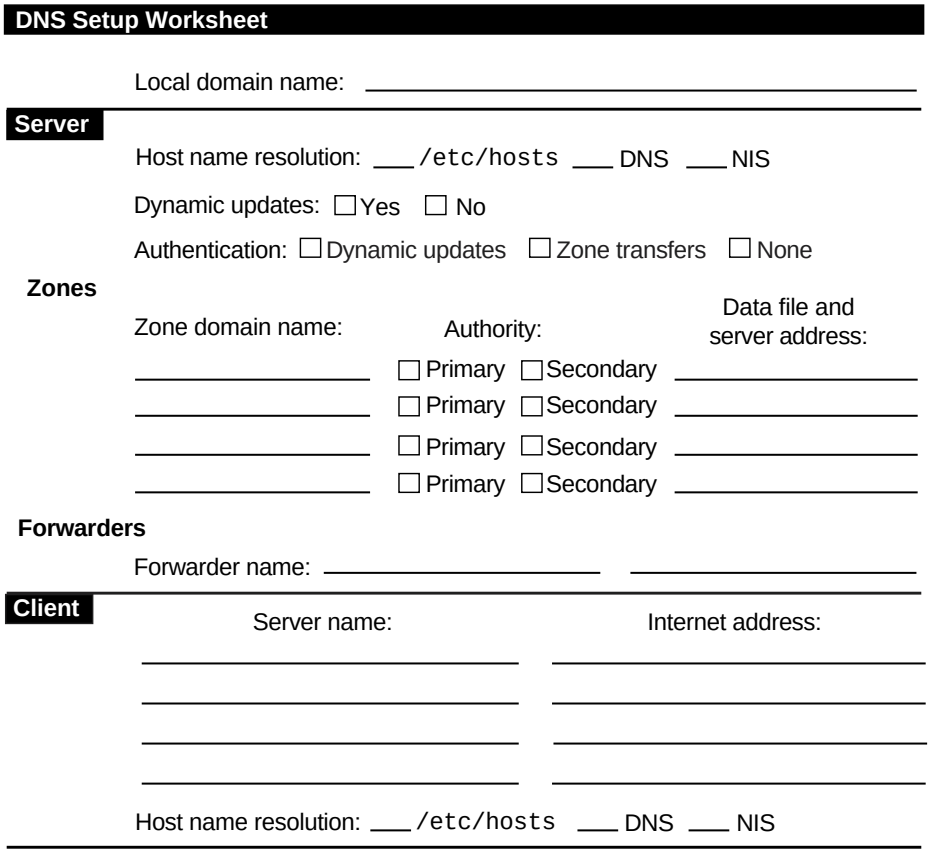
See Section 2.6 for information about configuring authentication for dynamic updates and zone transfers.

2.4 Planning DNS

Figure 2–3 shows the DNS Setup Worksheet, which you can use to record the information required to configure DNS. If you are viewing this manual online, you can use the print feature of your browser to print a copy of

this worksheet. The following sections explain the information you need to record on the worksheet.

Figure 2–3: DNS Setup Worksheet



DNS Setup Worksheet

Local domain name: _____

Server

Host name resolution: ___ /etc/hosts ___ DNS ___ NIS

Dynamic updates: ☐ Yes ☐ No

Authentication: ☐ Dynamic updates ☐ Zone transfers ☐ None

Zones

Zone domain name:	Authority:	Data file and server address:
_____	☐ Primary ☐ Secondary	_____
_____	☐ Primary ☐ Secondary	_____
_____	☐ Primary ☐ Secondary	_____
_____	☐ Primary ☐ Secondary	_____

Forwarders

Forwarder name: _____ _____

Client

Server name:	Internet address:
_____	_____
_____	_____
_____	_____
_____	_____

Host name resolution: ___ /etc/hosts ___ DNS ___ NIS

Local domain name

For a master server, the domain for which the server has primary authority. For client systems, the parent domain of which your local system is a part. For example, if your system's domain name is `cxcxcx.abc.xyz.com`, your local domain name is `abc.xyz.com`.

2.4.1 Server

Host name resolution

The order in which the local `/etc/hosts` file, DNS database, and NIS database are to be queried for host name resolution.

Indicate the order on the worksheet by placing the appropriate number next to each item. The following order is recommended:

1. Local hosts file
2. DNS database
3. NIS database

Dynamic updates

Check Yes if you want to enable dynamic client updates; otherwise, check No.

Authentication

If you want the master server to authenticate the DNS database updates it receives from clients, check Dynamic updates. If you want to authenticate zone transfers between master and slave servers, check Zone transfers. If you do not require authentication, check None.

If you plan to use the `nd6hostd` daemon to provide dynamic updates for IPv6 zones, do not enable authentication for these zones. The `nd6hostd` daemon does not support authentication.

Zone domain name

The name of the top-level domain in the zone.

Authority

If the server is a master server for the zone (maintains the zone database file), check Primary. If the server is a slave server for the zone (copies the zone database file from the master), check Secondary.

Data file and server address

For a master server, the full directory and file name specification for the file in which the master database of zone information will be stored.

For a slave server or stub server, the full directory and file name specification for the file in which a local copy of the database from the master server will be stored. Also, the IP address of the master server.

Forwarder name

The host name of a system or systems to which your server forwards queries that it cannot resolve locally. When the server receives a query that it cannot answer from its cache, it sends the query to a forwarder for resolution. If the forwarder cannot answer the query, the server might contact other servers directly. If your system is a Forward-only

server, you must include forwarder names; otherwise, forwarders are optional.

2.4.2 Client

Server name

The name of a server to contact for host name resolution. Specify up to three servers.

Internet address

A corresponding IP address for the server or servers.

Host name resolution

The order in which the local `/etc/hosts` file, DNS database, and NIS database are to be queried for host name resolution.

Indicate the order on the worksheet by placing the appropriate number next to each item. The following order is recommended:

1. Local hosts file
2. DNS database
3. NIS database

2.5 Configuring DNS

Use the SysMan Menu application of the Common Desktop Environment (CDE) Application Manager to configure DNS on servers and clients. To invoke the SysMan Menu application, follow the instructions in Section 1.2.1.

When you configure DNS, you must first set up the master server. You can configure the other systems in any order.

2.5.1 Configuring a Master Server

To configure a master server, do the following:

Note

If you are configuring an IPv6 master server, see Section 2.5.1.1 for more information.

1. Copy into the `/etc/namedb/src` directory the hosts file that you want to convert to the DNS hosts database.

To create a new file from which the hosts database will be created, you can update the master server's local `/etc/hosts` file (see *Network Administration: Connections*) and copy it into the `/etc/namedb/src` directory with the same `hosts` file name. If a system is in your DNS domain and is running DNS but is not included in the master server's hosts database, other systems in the domain cannot obtain its IP address.

2. From the SysMan Menu, select Networking→Additional Network Services→Domain Name Service (DNS(BIND))→Configure system as a DNS server to display the DNS Server Configuration dialog box.

   Alternatively, enter the following command on a command line:

   ```
   # /usr/bin/sysman dns_server
   ```

3. Enter your local domain name, the domain for which the master server will have primary authority, in the Local Domain field.
4. Select MASTER in the DNS Server Type pull-down menu.
5. Indicate the order in which to resolve host name queries in the Host Name Resolution Order field. Open the pull-down menu and choose from the list of options. Administrators usually use either the DNS Database, Local Host File, NIS option or the Local Host File, DNS Database, NIS option; the latter is recommended. Your choice is recorded in the `/etc/svc.conf` file.

   Alternatively, you can run the `svcsetup` script to customize service order selection. See Section 2.8.1 and `svcsetup`(8) for information about modifying the `svc.conf` file.
6. Select the Advanced DNS Options button to display the Advanced Options dialog box, which allows you to choose a different directory and files in which to save DNS server configuration information and cache data. Make changes, if necessary, then select OK to close the dialog box.
7. Select Next to display the Local Name Server list.

   a. Select Add to display the Add Name Server dialog box.

   b. Enter the host name and IP address for a name server that your master server will query for addresses it cannot resolve by itself.

   c. Select OK to accept the entry. If the server is not a known host, select Yes to add it to the `/etc/hosts` file.

      Repeat steps 7a through 7c, if necessary. It is best to specify two or three name servers.

   d. Select Next to accept the list of name servers. The addresses are later recorded in the `/etc/resolv.conf` file.

8. Select the appropriate check box to create the DNS database. Specify the name of the source file to use in the Hosts File field. Use `/etc/namedb/src/hosts` for the file you created in step 1 or the default for `/etc/hosts`. Select Next to continue.
9. Select the appropriate check box to start the `named` daemon and select Next to continue. The utility prompts you to change the host name of the system.
10. Select the appropriate check box to change the host name, if necessary. If you choose to change the host name, you are prompted to add `localhost` to the access control list. Select Yes to allow graphical user interfaces to be displayed properly on your newly renamed system.
11. Select Next to continue, then select Finish to save the configuration and start the `named` daemon.

    You are informed that you successfully configured the system as a DNS server. Select OK to close the DNS Server Configuration dialog box.

You can also modify your server configuration after the initial setup. See the online help for more information.

To enable dynamic updates on a DNS master server for IPv6 or Microsoft Windows network environments, see Section 2.5.1.2 and Section 2.6.

2.5.1.1 Configuring an IPv6 Master Server

Configuring an IPv6 master server is similar to configuring an IPv4 master server with a few exceptions. The following sections describe the exceptions.

2.5.1.1.1 DNS Configuration Files

The `/usr/examples/ipv6/namedb` directory contains DNS configuration files that show sample IPv6 information for you to study and adapt to your environment. Of the files in that directory, the following files contain IPv6 information that show reverse lookup addresses and dynamic update examples:

- `/usr/examples/ipv6/namedb/ipv6.rev`
- `/usr/examples/ipv6/namedb/ipv6.db`
- `/usr/examples/ipv6/namedb/named.conf`

After you customize these files for your environment, save the original files in the `/etc/namedb` directory, then move the customized files to that directory.

2.5.1.1.2 Server Guidelines

To configure a DNS server to operate in an IPv6 network environment, review the following guidelines:

- Select a node to function as an IPv6 name server.
- Dedicate a zone to IPv6 addresses or add IPv6 addresses to your enterprise's current zone.
- If you want global IPv6 name services, you must delegate a domain under the `ip6.int` domain for the reverse lookup of IPv6 addresses. Send mail to the following address to request a domain for reverse lookups:

  `bmanning@isi.edu`

  See RFC 1886 for more information.

  See *Network Administration: Connections* for information on how to create a reverse lookup zone name.
- If the system is already configured as a DNS server, change the `/etc/resolv.conf` file to point to the local node for name lookups, as follows:

  ```
  nameserver 127.0.0.1
  ```

2.5.1.2 Enabling Dynamic Updates to the DNS Database

To enable dynamic updates on a DNS master server for IPv6 or Microsoft Windows network environments, do the following:

Caution

Each time you reconfigure your DNS master server with the SysMan Menu, you must reenable dynamic updates by repeating these steps because your `named.conf` file will be rewritten.

1. Edit the `/etc/namedb/named.conf` file and add the `allow-update` substatement to the master zone statements (forward and reverse lookup) for which you want to enable dynamic updates, as follows:

   ```
   zone "zone-name" {
           type master;
           file "file-name";
           allow-update { any; };
   };

   zone  "rev-ip.in-addr.arpa" {
           type  master;
           file "file-name.rev";
           allow-update { any; };
   };
   ```

   For example, if you are enabling dynamic updates in an IPv6 zone, the zone statements might appear as follows after the change:

```
zone "ipv6.site1.corp.example" {
         type master;
         file "ipv6.site1.db";
         allow-update { any; };
};
zone "0.4.c.8.0.0.0.4.c.8.0.1.0.0.1.2.0.0.f.5.IP6.INT" {
        type master;
        file "ipv6.site1.rev";
        allow-update { any; };
};
```

Note that specifying any in the allow-update substatements allows any client to update the master DNS database. If you prefer to limit access to the database, see Section 2.6 for information about enabling authentication of dynamic updates. (However, note that the nd6hostd daemon on IPv6 clients does not support authentication.)

2. Start or restart the named daemon. See the online help for more information.

For information about configuring Microsoft Windows 2000 systems on a network with Tru64 UNIX DNS servers, see the Best Practice for *Integrating Windows 2000 DNS Clients with Tru64 UNIX DNS Services* on the Tru64 UNIX Publications Home Page at the following URL:

http://www.tru64unix.compaq.com/docs/

2.5.2 Configuring a Slave Server

To configure a slave server, invoke the SysMan Menu as documented in Section 1.2.1 and do the following:

1. From the SysMan Menu, select Networking→Additional Network Services→Domain Name Service (DNS(BIND))→Configure system as a DNS server to display the DNS Server Configuration dialog box.

   Alternatively, enter the following command on a command line:

   ```
   # /usr/bin/sysman dns_server
   ```

2. Enter your local domain name in the Local Domain field.
3. Select SLAVE in the DNS Server Type pull-down menu.
4. Indicate the order in which to resolve host name queries in the Host Name Resolution Order field. Open the pull-down menu and choose from the list of options. Administrators usually use either the DNS Database, Local Host File, NIS option or the Local Host File, DNS Database, NIS option; the latter is recommended. Your choice is recorded in the /etc/svc.conf file.

Alternatively, you can run the `svcsetup` script to customize service order selection. See Section 2.8.1 and `svcsetup`(8) for information about modifying the `svc.conf` file.

5. Select the Advanced DNS Options button to display the Advanced Options dialog box, which allows you to choose a different directory and files in which to save DNS server configuration information and cache data. Make changes, if necessary, then select OK to close the dialog box.
6. Select Next to display the Zones Served list.
   a. Select Add to display the Add Zone dialog box.
   b. Select the Slave radio button in the Authority field and enter the name of the zone (domain) for which this server will have secondary authority.
   c. Enter the name of the local file in which to store a copy of the database of zone information from the master server. Also, enter the IP address of the master server.
   d. Select OK to accept the entry. Repeat steps 6a through 6d for additional entries. At the very least, you must add a forward lookup entry and reverse lookup entry for each zone.

      Given the following `/etc/namedb/named.conf` file on the master server, you would add entries in the slave's Zones Served list for the *`domain.suffix`* and *`nn.nnn.in-addr.arpa`* zones:

      ```
      zone domain.suffix {
           type master;
           file "hosts.db";
           };

      zone nn.nnn.in-addr.arpa {
           type master;
           file "hosts.rev";
           };
      ```

7. Select Next to display the Local Name Server list.
   a. Select Add to display the Add Name Server dialog box.
   b. Enter the host name and IP address for a name server that your slave server will query for addresses that it cannot resolve by itself.
   c. Select OK to accept the entry. If the server is not a known host, select Yes to add it to the `/etc/hosts` file.

      Repeat steps 7a through 7c, if necessary. It is best to specify two or three name servers.
   d. Select Next to accept the list of name servers. The addresses are later recorded in the `/etc/resolv.conf` file.

8. Select the appropriate check box to start the `named` daemon and select Next to continue. The utility prompts you to change the host name of the system.
9. Select the appropriate check box to change the host name, if necessary. If you choose to change the host name, you are prompted to add `localhost` to the access control list. Select Yes to allow graphical user interfaces to be displayed properly on your newly renamed system.
10. Select Next to continue, then select Finish to save the configuration and start the `named` daemon.

    You are informed that you successfully configured the system as a DNS server. Select OK to close the DNS Server Configuration dialog box.

You can also modify your server configuration after the initial setup. See the online help for more information.

2.5.3 Configuring a Caching-Only Server

To configure a caching-only server, invoke the SysMan Menu as documented in Section 1.2.1 and do the following:

1. From the SysMan Menu, select Networking→Additional Network Services→Domain Name Service (DNS(BIND))→Configure system as a DNS server to display the DNS Server Configuration dialog box.

   Alternatively, enter the following command on a command line:

   ```
   # /usr/bin/sysman dns_server
   ```
2. Enter your local domain name in the Local Domain field.
3. Select CACHING in the DNS Server Type pull-down menu.
4. Indicate the order in which to resolve host name queries in the Host Name Resolution Order field. Open the pull-down menu and choose from the list of options. Administrators usually use either the DNS Database, Local Host File, NIS option or the Local Host File, DNS Database, NIS option; the latter is recommended. Your choice is recorded in the `/etc/svc.conf` file.

   Alternatively, you can run the `svcsetup` script to customize service order selection. See Section 2.8.1 and `svcsetup`(8) for information about modifying the `svc.conf` file.
5. Select the Advanced DNS Options button to display the Advanced Options dialog box, which allows you to choose a different directory and files in which to save DNS server configuration information and cache data. Make changes, if necessary, then select OK to close the dialog box.

6. Select Next to display the Local Name Server list.

   a. Select Add to display the Add Name Server dialog box.

   b. Enter the host name and IP address for a name server that your caching-only server will query for addresses that it cannot resolve by itself.

   c. Select OK to accept the entry. If the server is not a known host, select Yes to add it to the `/etc/hosts` file.

      Repeat steps 6a through 6c, if necessary. It is best to specify two or three name servers.

   d. Select Next to accept the list of name servers. The addresses are later recorded in the `/etc/resolv.conf` file.

7. Select the appropriate check box to start the `named` daemon and select Next to continue. The utility prompts you to change the host name of the system.

8. Select the appropriate check box to change the host name, if necessary. If you choose to change the host name, you are prompted to add `localhost` to the access control list. Select Yes to allow graphical user interfaces to be displayed properly on your newly renamed system.

9. Select Next to continue, then select Finish to save the configuration and start the `named` daemon.

   You are informed that you successfully configured the system as a DNS server. Select OK to close the DNS Server Configuration dialog box.

You can also modify your server configuration after the initial setup. See the online help for more information.

2.5.4 Configuring a Forward-Only Server

To configure a forward-only server, invoke the SysMan Menu as documented in Section 1.2.1 and do the following:

1. From the SysMan Menu, select Networking→Additional Network Services→Domain Name Service (DNS(BIND))→Configure system as a DNS server to display the DNS Server Configuration dialog box.

   Alternatively, enter the following command on a command line:

   ```
   # /usr/bin/sysman dns_server
   ```

2. Enter your local domain name in the Local Domain field.

3. Select FORWARDER in the DNS Server Type pull-down menu.

4. Indicate the order in which to resolve host name queries in the Host Name Resolution Order field. Open the pull-down menu and choose

from the list of options. Administrators usually use either the DNS Database, Local Host File, NIS option or the Local Host File, DNS Database, NIS option; the latter is recommended. Your choice is recorded in the `/etc/svc.conf` file.

Alternatively, you can run the `svcsetup` script to customize service order selection. See Section 2.8.1 and `svcsetup`(8) for information about modifying the `svc.conf` file.

5. Select the Advanced DNS Options button to display the Advanced Options dialog box, which allows you to choose a different directory and files in which to save DNS server configuration information and cache data. Make changes, if necessary, then select OK to close the dialog box.
6. Select Next to display the Forwarders list.
   a. Select the appropriate check box to indicate that you want to configure the system as a forward-only server.
   b. Select Add to display the Add Forwarder dialog box.
   c. Enter the IP address for a forwarder, a name server that your forward-only server will query for addresses on remote networks (like the Internet).
   d. Select OK to accept the entry. If the server is not a known host, select Yes to add it to the `/etc/hosts` file.

      Repeat steps 6b through 6d, if necessary. It is best to specify two or three forwarders.
   e. Select Next to accept the list of forwarders. The addresses are later recorded in the `/etc/resolv.conf` file.
7. Select Next to display the Local Name Server list.
   a. Select Add to display the Add Name Server dialog box.
   b. Enter the host name and IP address for a name server that your forward-only server will query for addresses on the local network.
   c. Select OK to accept the entry. If the server is not a known host, select Yes to add it to the `/etc/hosts` file.

      Repeat steps 7a through 7c, if necessary. It is best to specify two or three name servers.
   d. Select Next to accept the list of name servers. The addresses are later recorded in the `/etc/resolv.conf` file.
8. Select the appropriate check box to start the `named` daemon and select Next to continue. The utility prompts you to change the host name of the system.

9. Select the appropriate check box to change the host name, if necessary. If you choose to change the host name, you are prompted to add `localhost` to the access control list. Select Yes to allow graphical user interfaces to be displayed properly on your newly renamed system.
10. Select Next to continue, then select Finish to save the configuration and start the `named` daemon.

    You are informed that you successfully configured the system as a DNS server. Select OK to close the DNS Server Configuration dialog box.

You can also modify your server configuration after the initial setup. See the online help for more information.

2.5.5 Configuring a Stub Server

To configure a stub server, invoke the SysMan Menu as documented in Section 1.2.1 and do the following:

Note

When configuring stub service, run the SysMan Menu application on the server that will have authority for the subzone, not on the master server. See the definition for a stub server in Section 2.1 for more information.

1. From the SysMan Menu, select Networking→Additional Network Services→Domain Name Service (DNS(BIND))→Configure system as a DNS server to display the DNS Server Configuration dialog box.

   Alternatively, enter the following command on a command line:

   ```
   # /usr/bin/sysman dns_server
   ```
2. Enter your local domain name in the Local Domain field.
3. Select STUB in the DNS Server Type pull-down menu.
4. Indicate the order in which to resolve host name queries in the Host Name Resolution Order field. Open the pull-down menu and choose from the list of options. Administrators usually use either the DNS Database, Local Host File, NIS option or the Local Host File, DNS Database, NIS option; the latter is recommended. Your choice is recorded in the `/etc/svc.conf` file.

   Alternatively, you can run the `svcsetup` script to customize service order selection. See Section 2.8.1 and `svcsetup`(8) for information about modifying the `svc.conf` file.
5. Select the Advanced DNS Options button to display the Advanced Options dialog box, which allows you to choose a different directory and

files in which to save DNS server configuration information and cache data. Make changes, if necessary, then select OK to close the dialog box.

6. Select Next to display the Zones Served list.

   a. Select Add to display the Add Zone dialog box.

   b. Select the Stub radio button in the Authority field. Enter the name of the stub zone (domain).

   c. Enter the name of the local file in which to store a copy of the database of zone information from the master server. Also, enter the IP address of the master server.

   d. Select OK to accept the entry. Repeat steps 6a through 6d for additional entries. At the very least, you must add a forward lookup entry and a reverse lookup entry for each zone.

      Given the following `/etc/namedb/named.conf` file on the master server, you would add entries in the slave's Zones Served list for the *`domain.suffix`* and *`nn.nnn.in-addr.arpa`* zones:

```
zone domain.suffix {
     type master;
     file "hosts.db";
     };

zone nn.nnn.in-addr.arpa {
     type master;
     file "hosts.rev";
     };
```

7. Select Next to display the Local Name Server list.

   a. Select Add to display the Add Name Server dialog box.

   b. Enter the host name and IP address for a name server that your stub server will query for addresses that it cannot resolve by itself.

   c. Select OK to accept the entry. If the server is not a known host, select Yes to add it to the `/etc/hosts` file.

      Repeat steps 7a through 7c, if necessary. It is best to specify two or three name servers.

   d. Select Next to accept the list of name servers. The addresses are later recorded in the `/etc/resolv.conf` file.

8. Select the appropriate check box to start the `named` daemon and select Next to continue. The utility prompts you to change the host name of the system.

9. Select the appropriate check box to change the host name, if necessary. If you choose to change the host name, you are prompted to add

localhost to the access control list. Select Yes to allow graphical user interfaces to be displayed properly on your newly renamed system.

10. Select Next to continue, then select Finish to save the configuration and start the named daemon.

    You are informed that you successfully configured the system as a DNS server. Select OK to close the DNS Server Configuration dialog box.

You can also modify your server configuration after the initial setup. See the online help for more information.

2.5.6 Configuring a DNS Client

To configure a DNS client, invoke the SysMan Menu as documented in Section 1.2.1 and do the following:

1. From the SysMan Menu, select Networking→Additional Network Services→Domain Name Service (DNS(BIND))→Configure system as a DNS client to display the Configure DNS Client dialog box.

   Alternatively, enter the following command on a command line:

   ```
   # /usr/bin/sysman dns_client
   ```

2. Enter your local domain name in the Local Domain field.
3. Select Add to add a name server.
4. Enter the host name and the IP address for the name server.

   The addresses are recorded in the /etc/resolv.conf file, where the resolver uses them to determine the IP addresses of name servers it will query.

5. Select OK to add the host name to the list of name servers. If the specified host is not listed in the /etc/hosts file, the script prompts you to add it to that file. Select Yes or No.

   To add other name servers, repeat steps 3 through 5. You can specify up to three name servers.

6. Indicate the order in which to resolve host name queries in the Host Name Resolution Order field. Open the pull-down menu and choose from the list of options. Administrators usually use either the DNS Database, Local Host File, NIS option or the Local Host File, DNS Database, NIS option; the latter is recommended. Your choice is recorded in the /etc/svc.conf file.

   Alternatively, you can run the svcsetup script to customize service order selection. See Section 2.8.1 and svcsetup(8) for information about modifying the svc.conf file.

7. Configure your system to search alternate domains for address resolution, if necessary.

   If you enter domains to search in this dialog box, you must include your local domain, otherwise, DNS will not search it. DNS searches the domains in the order that you specify; therefore, it is best to specify the local domain as the first entry in the list.

   If you do not enter domains to search, DNS searches your local domain by default.

   To specify a list of domains to search:

   a. Select Domains Searched to display the associated dialog box.
   b. Select Add to display the Add/Modify dialog box.
   c. Enter the name of a domain to search.
   d. Select OK to accept the entry. Repeat steps 7b through 7d, if necessary. You can specify up to six domains.
   e. Select OK to accept the list of domains to be searched.

8. Select OK to accept the configuration. The script prompts you to change the host name of the system.
9. Select Yes or No as appropriate. If you choose Yes to change the host name, you are prompted to add `localhost` to the access control list. Select Yes to allow graphical user interfaces to be displayed properly on your newly renamed system.
10. Select OK to close the Configure DNS Client dialog box.

You can also modify your client configuration after the initial setup. See the online help for more information.

For IPv6 network environments, to enable dynamic updates for a DNS client, do the following:

1. Run the `ip6_setup` script and enable dynamic updates of IPv6 addresses by entering **y** at the prompt. Provide a fully qualified domain name for the IPv6 host. See *Network Administration: Connections* for more information.
2. Configure the DNS/BIND server to allow the updates (see Section 2.5.1.2).

If you do not want to enable dynamic updates, you must still run the `ip6_setup` script, but no special DNS configuration is necessary.

2.6 Configuring Authentication

The following sections describe how to configure authentication on DNS servers for the following purposes:

- Secure dynamic updates
- Secure zone transfers

Authentication is useful only when the private key remains a secret between the servers; therefore, it is prudent to change this key frequently and save the key file as specified in the following sections to prevent the key from being compromised.

2.6.1 Configuring Secure Dynamic Updates

If you plan to use the `nd6hostd` daemon to dynamically update IPv6 zones, do not enable authentication for these zones. The `nd6hostd` daemon does not support authentication.

To configure a master server to authenticate dynamic updates it receives from new DNS clients (Microsoft Windows systems), do the following:

Caution

Each time you reconfigure your DNS master server with the SysMan Menu, you must reenable secure client updates by repeating these steps because your `named.conf` file will be rewritten.

1. Generate a private key using the `dnskeygen` command, as follows:

   # **dnskeygen -H *size* -h -c -n *key-name***

   Valid key sizes are 512, 576, 640, 704, 768, 832, 896, 960, and 1024 bytes. Larger keys are more cumbersome, but they are more secure.

   You can supply any name for a key, but it is best to give the keys canonical names so they are easy to distinguish. For example, if hosts from the xyz.corp.com zone send dynamic updates to your master server, marlin.xyz.corp.com, you might want to name your key xyznet-marlin_update.

   The `dnskeygen` command produces two files:

   - `K<`*key-name*`><`*proto-id*`><`*key-id*`>.key`
   - `K<`*key-name*`><`*proto-id*`><`*key-id*`>.private`

   Hereafter, these files are referred to as the `.key` and `.private` files.

   For more information about generating keys, see `dnskeygen`(1).

2. Create a file to contain the key configuration statement for the update. This file must be read/writeable only by superuser to prevent the private key from being compromised. For example:

```
# cd /etc/namedb
# touch key-config-file
# chmod 600 key-config-file
```

Although it is not necessary, you might want to call the *key-config-file* named.keys.

3. Incorporate the key information from the .private file into the *key-config-file* by adding the following key statement:

```
key key-name {
    algorithm hmac-md5;
    secret "generated-key";
};
```

In the key statement, replace *key-name* with the name of the key and replace *generated-key* with the entire private key as it appears in the .private file. It is best to enter the key by opening the .private file in another window, copying the necessary key text, and pasting the text into the text editor window. There must be no line feeds or spaces between the quotes that contain the key; if even one character is entered incorrectly, authentication fails.

4. Add the following include statement to the top of the /etc/namedb/named.conf file:

```
include "/etc/namedb/key-config-file";
```

Replace *key-config-file* with the name of the key configuration file you created in steps 2 and 3.

When the named daemon starts and reads the DNS data file, it calls the *key-config-file* and parses its contents.

5. Enable secure dynamic updates for the master zone by adding the allow-update substatement to the master zone statements (for forward and reverse lookups) in the named.conf file :

```
zone "zone-name" {
        type master;
        file "file-name";
        allow-update {
             key key-name;
        };
};

zone  "rev-ip.in-addr.arpa" {
        type  master;
        file "hosts.rev";
        allow-update {
```

```
            key key-name;
        };
};
```

Replace *key-name* with the name of the file you created in steps 2 and 3.

Specifying a key in this statement ensures that updates are successful only if they are signed with the private key.

6. Restart the named daemon by issuing the following command:

```
# /sbin/init.d/named restart
```

After you configure the master server to support secure dynamic DNS updates from new hosts, you can distribute the private key as necessary to administrators who need to add these hosts to the network. It is best to physically distribute the key on magnetic or optical media as opposed to sending it over the network where it can be compromised.

You can format a floppy disk for this purpose. See mtools(1) for information about formatting and reading Microsoft Windows-compatible floppy disks on a Tru64 UNIX system. If the described tools are not available, you need to install the OSFDOSTOOLS subset.

For information about configuring Microsoft Windows 2000 systems on a network with Tru64 UNIX DNS servers, see the Best Practice for *Integrating Windows 2000 DNS Clients with Tru64 UNIX DNS Services* on the Tru64 UNIX Publications Home Page at the following URL:

http://www.tru64unix.compaq.com/docs/

Note that when clients send updates to the master server, the named daemon does not immediately update the master database files. It creates temporary *database*.ixfr and *database*.log files where it logs the changes until they can be incorporated into the database. However, almost immediately, the daemon becomes aware of the updates in memory. You can verify them with the nslookup command as explained in Section 2.8.3.1.

2.6.2 Configuring Secure Zone Transfers

To configure a master server and slave servers to use authentication for zone transfers, do the following:

Caution

Each time you reconfigure DNS with the SysMan Menu, you must re-enable secure zone transfers by repeating these steps because your named.conf file will be rewritten.

1. On the master server, perform steps 1–4 as specified in Section 2.6.1.

When creating a key name, choose a name that describes the zone transfer. For example, if the master server, marlin.xyz.corp.com, is sending updates to the slave server, minnow.xyz.corp.com, for the xyz.corp.com zone, you might name the key xyznet-marlin-minnow_transfer.

2. On the master server, add the `allow-transfer` substatement to the master `zone` statements (for forward and reverse lookups) in the `/etc/namedb/named.conf` file:

```
include "/etc/namedb/key-file";
             .
             .
             .
zone "zone-name" {
        type master;
        file "hosts.db";
        allow-transfer {
             key key-name;
    };

zone  "rev-ip.in-addr.arpa" {
        type  master;
        file  "hosts.rev";
        allow-transfer {
             key key-name;
        };

};
```

   Replace *key-name* with the name of the key as you specified it in the key configuration file you created in steps 2 and 3 of Section 2.6.1.

   Adding this server statement ensures that the master server transfers the zone only if the request is signed with the private key. It also ensures that the master server signs the zone transfer with the key before it sends the data to the slave server.

3. Transfer the key configuration file (*key-config-file* or `named.keys`) from the master server to the slave server(s). It is best to physically transfer this file on magnetic or optical media as opposed to sending it over the network where it can be compromised.

   You can format a floppy disk for this purpose. See `mtools`(1) for information about formatting and reading Microsoft Windows-compatible floppy disks on a Tru64 UNIX system. If the described tools are not available, you need to install the `OSFDOSTOOLS` subset.

   On the slave server(s), ensure that the permissions are set for read/writable only by superuser:

```
# chmod 600 key-config-file
```

4. On the slave server(s), add an `include` statement to the `named.conf` file to call the *key-config-file*. Also, insert the `server` statement after the `include` statement and before any `zone` statements:

```
include "/etc/namedb/key-config-file";
        .
        .
        .
server ip-address {
        keys {key-name};
};
```

   Replace *key-config-file* with the name of the key configuration file you copied from the master server. Replace *ip-address* with the IP address of the master server. Finally, replace *key-name* with the name of the key you specifed in the *key-config-file*.

   Adding the `server` statement ensures that the slave server signs requests for zone transfers from the master server with the private key. It also ensures that the slave server authenticates signed zone transfers from the master server before it incorporates them into its data files.

5. Restart the `named` daemon on the master server and the slave server(s) by issuing the following command:

```
# /sbin/init.d/named restart
```

2.6.3 Authentication Example

The following examples show sample `named.keys` and `named.conf` files that implement both secure dynamic updates and secure zone transfers. These configuration files describe a network in which there is a DNS master server called marlin.ocean.corp.com and a slave server called minnow.ocean.corp.com.

Example 2–1: Sample named.keys File for Authentication

```
key oceannet-client_update { 1
        algorithm hmac-md5; 2
        secret "lSYbJjbTOLH2DB+kRpf0fcTJk0mOca90GDGdn5R7L2vPhyCx
daGhHp0o2pDU+PSzclE3Yk6Xg8jOkpRExx+2yw=="; 3
};

key oceannet-marlin-minnow_transfer { 4
        algorithm hmac-md5;
        secret "648NyJi33LMhf00iavHjbkgqcTMJ7lZD4/r0DF9wgIQ2WH2b
peHLYjz2qYMrx1dMYw9E9gDp6F6LTMDHHCvFlw==";
```

In Example 2–1, the lines serve the following purpose:

1 Defines the `oceannet-client_update` key, which will be used for secure dynamic updates from clients in the ocean.corp.com zone.

2 Specifies the encryption algorithm. Keys for dynamic updates and zone transfers must be `hmac-md5`.

3 Specifies the key string. This string must contain no spaces or carriage returns.

4 Defines the `oceannet-marlin-minnow_transfer` key, which will be used for zone transfers between the master server, marlin.ocean.corp.com, and the slave server, minnow.ocean.corp.com.

Example 2–2: Sample Master Server named.conf File for Authentication

```
include "/etc/namedb/named.keys"; [1]

options {
        directory  "/etc/namedb/";
};

zone  "0.0.127.in-addr.arpa" {
        type  master;
        file  "named.local";
};

zone  "ocean.corp.com" {
        type  master;
        file  "hosts.db";
        allow-update {
                key oceannet-client_update; [2]
        };
        allow-transfer {
                key oceannet-marlin-minnow_transfer; [3]
        };
};

zone  "6.134.20.in-addr.arpa" {
        type  master;
        file  "hosts.rev";
        allow-update {
                key oceannet-client_update; [2]
        };
        allow-transfer {
                key oceannet-marlin-minnow_transfer; [3]
        };
};

zone  "." {
```

Example 2–2: Sample Master Server named.conf File for Authentication (cont.)

```
        type  hint;
        file  "named.ca";
};
```

In Example 2–2, the lines serve the following purpose:

1. Calls the aforementioned `named.keys` file into the `named.conf` file.
2. These lines specify that dynamic updates for the `ocean.corp.com` zone must be authenticated with the `oceannet-client_update` key before they are incorporated into the DNS database.
3. These lines specify that zone transfer requests for the `ocean.corp.com` zone must be authenticated with the `oceannet-marlin-minnow_transfer` key before any data is sent to the slave server(s).

Example 2–3: Sample Slave Server named.conf File for Authentication

```
include "/etc/namedb/named.keys";

server 20.134.6.2 {
        keys { oceannet-marlin-minnow_transfer }; [1]
};

options {
        directory  "/etc/namedb/";
};

zone  "0.0.127.in-addr.arpa" {
        type  master;
        file  "named.local";
};

zone  "pubs.zk3.dec.com" {
        type  slave;
        file  "/etc/namedb/hosts.db";
        masters  {
                20.134.6.2; [2]
        };
};

zone  "6.134.20.in-addr.arpa" {
        type  slave;
        file  "/etc/namedb/hosts.rev";
        masters  {
```

Example 2–3: Sample Slave Server named.conf File for Authentication (cont.)

```
                20.134.6.2; [2]
        };
};

zone  "." {
        type  hint;
        file  "named.ca";
};
```

In Example 2–3, the lines serve the following purpose:

[1] Specifies that the slave server is to use the `oceannet-marlin-minnow_transfer` key for authentication of all communication between itself and `20.134.6.2` (marlin.ocean.corp.com).

[2] These lines specify that `20.134.6.2` is the master server for the `ocean.corp.com` zone, and that it will provide the authoritative data for that zone.

For more information about the statements in the `named.conf` file, see `named.conf`(4) and the *Bind Configuration File Guide* on the Tru64 UNIX Documentation CD-ROM.

2.7 Deconfiguring DNS

Use the SysMan Menu application of the Common Desktop Environment (CDE) Application Manager to deconfigure DNS servers and clients. To invoke the SysMan Menu application, follow the instructions in Section 1.2.1.

When you deconfigure DNS, the service stops and the DNS server and client configuration information is deleted from the system. This action cannot be undone. To restore DNS, you must configure it again using the SysMan Menu.

To deconfigure DNS, do the following:

1. From the SysMan Menu, select Networking→Additional Network Services→Domain Name Service (DNS(BIND))→Deconfigure DNS on this system to display the Deconfigure DNS dialog box.

   Alternatively, enter the following command on a command line:

   # **/usr/bin/sysman dns_deconfigure**

2. Select Yes to deconfigure DNS on the system.
3. Select OK to close the Deconfigure DNS dialog box.

2.8 Managing DNS Servers and Clients

This section describes how to perform the following DNS tasks:

- Modify the `svc.conf` file with the `svcsetup` script
- Update DNS data files on the master server
- Obtain host name and IP address information

2.8.1 Modifying the svc.conf File with svcsetup

You can modify the `/etc/svc.conf` file without running the DNS Configuration application. To do this, you invoke the `svcsetup` script using the following command:

```
# /usr/sbin/svcsetup
```

Once invoked, use the following steps to edit the `/etc/svc.conf` file:

1. Press the Return key following the informational messages to continue.
2. Press the Return key to choose the `m` option from the Configuration Menu.
3. Choose option 2 from the Change Menu. Option 2 corresponds to the `hosts` database.
4. Enter the number that corresponds to the order in which you want the services running on your system queried for `hosts` data.

   Listing local first means that the local `/etc/hosts` file is searched first for the requested information. If the information is not found locally, then DNS servers, NIS servers, or both, are queried, depending on which options you choose.

   Note

   For better performance, it is best if the first service that your system queries for all databases is local, regardless of what services you are running.

   Choose option 3, 4, 5, or 6 to configure the `svc.conf` file so that DNS serves `hosts` information.

   The `svcsetup` script indicates that it is updating the `/etc/svc.conf` file. When the script is finished updating the file, it notifies you and the system prompt (#) is displayed.

2.8.2 Updating DNS Data Files on the Master Server

If you have not configured dynamic updates, as discussed in Section 2.2 and Section 2.5.1.2, you will need to manually update the DNS data files when you connect new hosts to the network.

To add a new host, follow these steps:

1. Edit the `/etc/namedb/src/hosts` file to add the new host.
2. Change to the `/etc/namedb` directory and enter one of the following commands:

```
# make hosts
# make all
```

After you edit the `hosts` file and enter the `make` command, the DNS conversion scripts (which are in the `/etc/namedb/bin` directory) do the following for you:

1. Create the new hosts databases: `hosts.db` and `hosts.rev`.
2. Place the new databases in the `/etc/namedb` directory.
3. Send a signal to the `named` daemon to reload all databases that have changed.

Note

If you have manually entered mail exchanger (MX) records in the `named.local` file, these records are lost. You will have to edit the `named.local` file and add the MX records.

The DNS database conversion scripts also increment the serial number field of the start of authority (SOA) entry in the database file and inform the slave servers that it is time to refresh their data.

The process is the same for all of the valid files in the master server's `/etc/namedb/src` directory. Scripts are provided to create the `named.local` and `named.ca` databases.

2.8.3 Obtaining Host Name and IP Address Information

There are several ways that you can obtain information about host names, IP addresses, and user information from a system using DNS. The following sections provide an introduction to two commands: `nslookup` and `whois`.

2.8.3.1 The nslookup Command

You can use the `nslookup` command to noninteractively and interactively query DNS for information about hosts on local and remote domains. You can also find information about DNS resource records such as mail exchanger (MX), name server (NS), and so forth.

For a noninteractive query, use the following syntax:

nslookup *hostname*

The output is the server name and address and the host name and address.

For an interactive query, use the following syntax:

nslookup

The output is the default server name and address and the `nslookup` prompt, a greater than sign (>).

For example, to obtain information about MX, you need to query `nslookup` interactively, supplying a valid domain name. The following example shows how to find the mail recipient for the domain `corp.com`:

```
# nslookup
Default Server:  localhost
Address:  127.0.0.1

> set querytype=mx
> corp.com
Server:  localhost
Address:  127.0.0.1
findmx.corp.com      preference = 100, mail exchanger = gateway.corp.com
gateway.corp.com     inet address = 128.54.54.79
> Ctrl/D
#
```

A good way to learn how to use the `nslookup` command is to experiment with it. To obtain a list of the interactive `nslookup` command options, enter a question mark (?) at the `nslookup` prompt. For further information, see `nslookup`(1).

For a detailed description of the many different types of DNS resource records, see Appendix G.

2.8.3.2 NIC whois Service

The Network Information Center (NIC) `whois` service allows you to access the following information about a domain:

- The name of the domain
- The name and address of the organization responsible for the domain
- The domain's administrative, technical, and zone contacts

- The host names and network addresses of sites providing DNS for the domain
- The registered users in the domain

For example, to use the NIC `whois` service to obtain information about a domain named `hp.com`, use the `whois` command and specify the domain name as follows:

```
# whois hp.com
Whois Server Version 1.3

Domain names in the .com, .net, and .org domains can now be registered
with many different competing registrars. Go to http://www.internic.net
for detailed information.

   Domain Name: HP.COM
   Registrar: NETWORK SOLUTIONS, INC.
   Whois Server: whois.networksolutions.com
   Referral URL: http://www.networksolutions.com
   Name Server: ATLREL1.HP.COM
   Name Server: ATLREL2.HP.COM
   Name Server: HPLB.HPL.HP.COM
   Name Server: PALREL1.HP.COM
   Name Server: PALREL2.HP.COM
   Updated Date: 26-mar-2002


>>> Last update of whois database: Mon, 15 Jul 2002 04:49:51 EDT <<<

The Registry database contains ONLY .COM, .NET, .ORG, .EDU domains and
Registrars.
```

According to the output, the `hp.com` domain is registered with Network Solutions, Inc. To find out more about the `hp.com` domain, you can use the `-h` option to query the `whois` server at Network Solutions specifically, as follows:

```
# whois -h whois.networksolutions.com hp.com
.
.

Registrant:
Hewlett-Packard Company (HP-DOM)
   3000 Hanover Street
   Palo Alto, CA 94304
   US

   Domain Name: HP.COM

   Administrative Contact, Technical Contact:
      HP Hostmaster  (HH15-ORG)         hostmaster@HP.COM
      Hewlett-Packard Company
      3404 East Harmony Rd., MS 68
      Fort Collins, CO 80528
      U.S.A.
      800-524-7638
      Fax- 970-898-2836

   Record expires on 04-Mar-2003.
   Record created on 03-Mar-1986.
   Database last updated on 15-Jul-2002 14:02:33 EDT.
```

```
Domain servers in listed order:

PALREL1.HP.COM              156.153.255.242
ATLREL1.HP.COM              156.153.255.210
PALREL2.HP.COM              156.153.255.234
ATLREL2.HP.COM              156.153.255.202
HPLB.HPL.HP.COM             192.6.10.2
```

3

Network Information Service

The Network Information Service (NIS, formerly Yellow Pages) is a distributed data lookup service for sharing information on a local area network (LAN). NIS allows you to coordinate the distribution of database information throughout your networked environment.

This chapter describes:

- The NIS environment (Section 3.1)
- How to plan for your NIS configuration (Section 3.2)
- How to configure your system for NIS (Section 3.3)
- How to manage an NIS server (Section 3.4)
- How to manage an NIS client (Section 3.5)

For introductory information on NIS, see `nis_intro`(7). For troubleshooting information, see Section 9.6 for clients and Section 9.5 for servers.

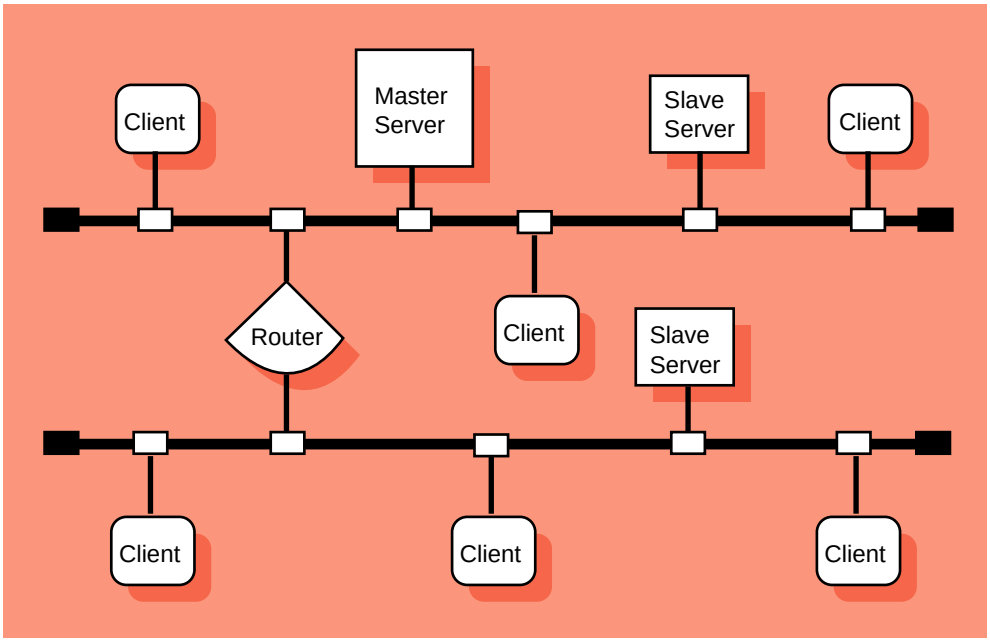
3.1 NIS Environment

In an NIS environment, systems can have the following roles:

- Master server — A system that stores the master copy of the NIS database files, or maps, for the domain in the `/var/yp/`*`DOMAIN`* directory and propagates them at regular intervals to the slave servers. Only the master maps can be modified. Each domain can have only one master server.
- Slave server — A system that obtains and stores copies of the master server's NIS maps. These maps are updated periodically over the network. If the master server is unavailable, the slave servers continue to make the NIS maps available to clients. Each domain can have multiple slave servers distributed throughout the network.
- Client — Any system that queries NIS servers for NIS database information. Clients do not store and maintain copies of the NIS maps locally for their domain.

Figure 3–1 shows a domain in which there is a master server, two slave servers, and some clients.

Figure 3–1: NIS Configuration



ZK-1145U-AI

By default, NIS distributes the `aliases` (`mail.aliases`), `group`, `hosts`, `netgroup`, `networks`, `passwd`, `protocols`, `rpc`, and `services` databases. (The `mail.aliases` and `netgroup` databases are created exclusively for NIS.) You can also create and distribute the enhanced security extended profile database, and site-specific customized databases, such as NFS Automount and AutoFS maps.

To configure NIS with support for enhanced security, and optionally create secure versions of NIS maps, carefully read the instructions in the *Security Administration* guide before proceeding with the setup described in this chapter. For information on creating Automount and AutoFS maps for distribution by NIS, see Appendix A. For information on creating and distributing other site-specific NIS maps, see the Section 3.4.6.

3.2 Planning NIS

This section describes the tasks you must complete before configuring NIS.

3.2.1 Verifying That the Additional Networking Services Subset is Installed

For NIS servers, verify that the Additional Networking Services subset is installed by entering the following command:

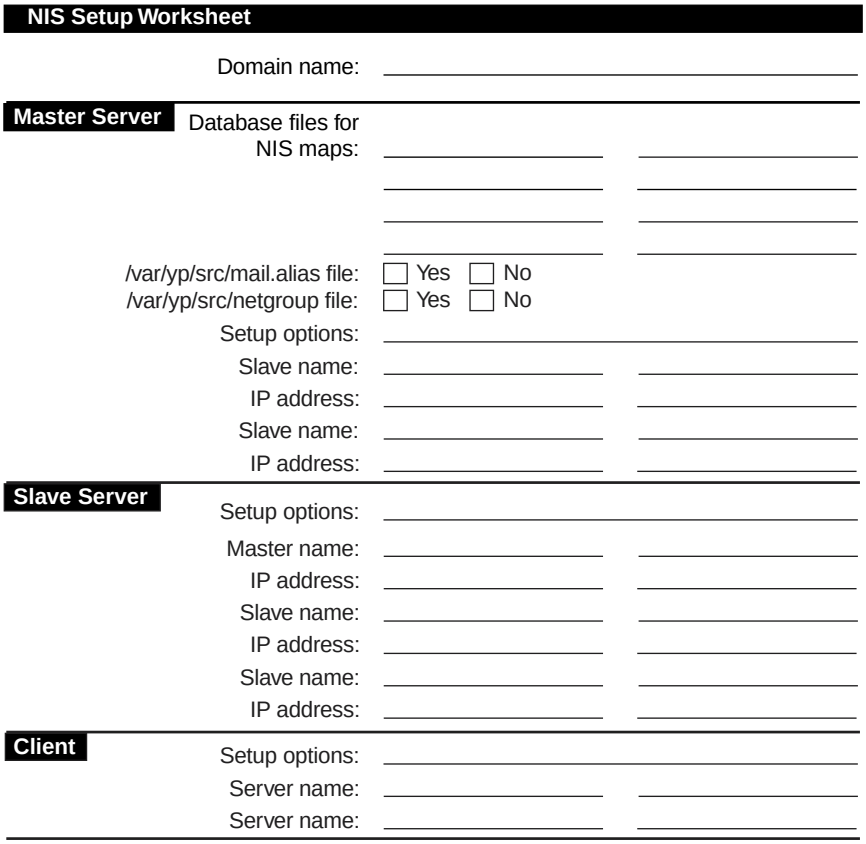
```
# setld –i | grep OSFINET
```

If the subset is not installed, install it by using the `setld` command. For more information on installing subsets, see `setld`(8) or the *Installation Guide*.

3.2.2 Preparing for the Configuration

Figure 3–2 shows the NIS Setup Worksheet, which you can use to record the information required to configure NIS. If you are viewing this manual online, you can use the print feature to print a copy of this worksheet. The following sections explain the information you need to record on the worksheet.

Figure 3–2: NIS Setup Worksheet



NIS Setup Worksheet

Domain name: _____

Master Server

Database files for NIS maps: _____ _____

_____ _____

_____ _____

_____ _____

/var/yp/src/mail.alias file: ☐ Yes ☐ No

/var/yp/src/netgroup file: ☐ Yes ☐ No

Setup options: _____

Slave name: _____ _____

IP address: _____ _____

Slave name: _____ _____

IP address: _____ _____

Slave Server

Setup options: _____

Master name: _____ _____

IP address: _____ _____

Slave name: _____ _____

IP address: _____ _____

Slave name: _____ _____

IP address: _____ _____

Client

Setup options: _____

Server name: _____ _____

Server name: _____ _____

Domain name

The domain name (1 to 31 alphanumeric characters). All systems in the domain must declare the same domain name.

An NIS domain is an administrative entity that consists of a master server, one or more slave servers, and numerous clients. All systems in a domain share the same set of NIS database files.

Note

An NIS domain name is not the same as a DNS domain name. Furthermore, an NIS domain name is case-sensitive. Be very careful when specifying it. If you configure the system with an incorrect NIS domain name, all NIS-related operations (such as logging in and `ls -l` commands) hang for several minutes, then fail.

NIS runs on each system in your network. You must decide what role each system will play within the NIS domain that you are creating. Select one host to be the master server; there can be only one master server for each domain. Select one or more hosts to be slave servers. The rest of the hosts can run as NIS clients. (The master server and all slave servers are also considered to be NIS clients.)

Once you have determined a role for each system, fill in the remainder of the worksheet as specified in the following sections.

3.2.2.1 Master Server

Database files for NIS maps

The files you want to make into NIS maps. Choose from the following list:

- `/etc/group`
- `/etc/hosts`
- `/etc/networks`
- `/etc/passwd`
- `/etc/protocols`
- `/etc/rpc`
- `/etc/services`

/var/yp/src/mail.aliases file

The `mail.aliases` file, which is based on the `/var/adm/sendmail/aliases` file, defines network-wide mail aliases. If you want to define and distribute mail aliases on your network, check Yes; otherwise, check No.

If you choose not to create a `mail.aliases` file, the `nissetup` script issues an informational message that it cannot find the `mail.aliases` file while it is building the NIS maps. For information on defining mail aliases, see `aliases`(4).

/var/yp/src/netgroup file

The `netgroup` file defines network-wide groups and is used for permission checking when doing remote mounts, remote logins, and remote shells. If you want to define and distribute `netgroup` information on your network, check Yes; otherwise, check No.

If you choose not to create a `netgroup` file, the `nissetup` script issues an informational message that it cannot find the `netgroup` file while

it is building the NIS maps. For information on defining network groups, see `netgroup`(4).

Setup options

The list of setup options for master servers follows. Write the options you want to use in the appropriate place in the worksheet.

- Run the `yppasswdd` daemon.

  The `yppasswdd` daemon allows users to update their passwords in the master copy of the password file by issuing the `yppasswd` command on any system in the NIS domain. If you want users to be able to update their NIS-distributed passwords without administrator intervention, run the `yppasswdd` daemon.

  The `yppasswdd` daemon runs only on the master server.

- Create base or enhanced security versions of the NIS maps.

  Tru64 UNIX security can be configured in either base or enhanced authentication mode. Enhanced security includes an additional `prpasswd` map that contains extended user profile information. Before configuring NIS to distribute this `prpasswd` map, see *Security Administration*, which describes important operational differences and additional steps necessary for NIS configuration in a secure environment.

- Create NIS maps in btree format.

  If you serve very large maps, you might want to have NIS maintain these maps as btree files, which significantly reduces the time required to build and push very large maps. However, the use of btree files might degrade performance slightly for relatively small maps. See `btree`(3) for more information about the btree format.

  If you intend to use enhanced security with NIS, it is best to maintain your maps in btree format.

- Run the `ypbind` daemon with the –`s` option, which requires the server to use a reserved port.

  For security purposes, it is best to run NIS with the –`s` option.

- Lock the `ypbind` daemon to a particular domain name and server list by specifying the `-S` option.

  Normally, hosts broadcast NIS requests on the network and the first available server answers the request. The –`S` option allows you to lock the `ypbind` daemon to a particular domain and set of servers. Requests are made directly to the specified servers, rather than being broadcast. For security purposes, it is best to run NIS with the –`S` option.

If you choose to run NIS with the –S option, you must know the host names and IP addresses of the servers to which you are locking the ypbind daemon. You will add them to the local hosts file during configuration.

Security Note

When using the nissetup script to set up an NIS server that is running with enhanced security, you must answer Yes to the question about locking the domain name and authorized servers (the ypbind –S option). For a master server, the server is bound to itself by default.

- Run NIS with the –ypset option or the –ypsetme option.

  The –ypset option allows a user logged in as root on any system in your domain to bind your system to a particular server. The –ypsetme option allows ypbind to accept –ypset requests only from the local system. For security purposes, it is best to disallow all ypset requests.

- Create and distribute Automount or AutoFS maps.

  The automount and autofsd daemons, which are alternatives to mounting remote file systems in the /etc/fstab file, allow users to mount remote file systems on an as-needed basis. When you use NIS to distribute the maps for these daemons, you create the maps on the NIS master server and distribute them to NIS slave servers and clients. For information on creating these maps, see Appendix A. For information on administering the maps, see Section 4.1.2.

  Whether or not you use Automount or AutoFS depends on your site's networking environment.

Slave name

The name of each slave server in the domain.

IP address

The IP address of each slave server in the domain.

3.2.2.2 Slave Server

Setup options

The list of setup options for slave servers follows. Write the options you want to use in the appropriate place in the worksheet.

- Maintain base or enhanced security versions of the NIS maps.

  Tru64 UNIX security can be configured in either base or enhanced authentication mode. Enhanced security includes an additional `prpasswd` map that contains extended user profile information. Before configuring NIS to distribute this `prpasswd` map, see *Security Administration*, which describes important operational differences and additional steps necessary for NIS configuration in a secure environment.

- Maintain NIS maps in btree format.

  If you serve very large maps, you might want NIS to maintain these maps as btree files, which significantly reduces the time required to push very large maps. However, it might degrade performance slightly for relatively small maps. See `btree`(3) for more information about the btree format.

  If you intend to use enhanced security with NIS, it is best to maintain your maps in btree format.

- Run the `ypbind` daemon with the –`s` option, which requires the server to use a reserved port.

  For security purposes, it is best to run NIS with the –`s` option.

- Lock the `ypbind` daemon to a particular domain name and server list by using the `-S` option.

  Normally, hosts broadcast NIS requests on the network and the first available server answers the request. The –`S` option allows you to lock the `ypbind` daemon to a particular domain and set of servers. Requests are made directly to the specified servers, rather than being broadcast. For security purposes, it is best to run NIS with the –`S` option.

  If you choose to run NIS with the –`S` option, you must know the host names and IP addresses of the servers to which you are locking the `ypbind` daemon to successfully complete the configuration process.

  Security Note

  When using the `nissetup` script to set up an NIS server that is running with enhanced security, you must answer Yes to the question about locking the domain name and authorized servers (the `ypbind –S` option). For a slave server, the server is bound to itself by default and optionally to the master server and any other slave servers.

- Run NIS with the –`ypset` option or the –`ypsetme` option.

The –ypset option allows a user logged in as root on any system in your domain to bind your system to a particular server. The –ypsetme option allows ypbind to accept –ypset requests only from the local system. For security purposes, it is best to disallow all ypset requests.

- Distribute Automount or AutoFS maps.

  The automount and autofsd daemons, which are alternatives to mounting remote file systems in the /etc/fstab file, allow users to mount remote file systems on an as-needed basis. When you use NIS to distribute the maps for these daemons, you can configure the slave server to receive the maps from the master server, distribute them to clients, and use them to mount remote file systems. For information on creating these maps, see Appendix A. For information on administering the maps, see Section 4.1.2.

  Whether or not you use Automount or AutoFS depends on your site's networking environment.

Master name

The host name of the master server in your domain.

IP address

The IP address of the master server in your domain.

Slave name

The name of another slave server in your domain. Specify several servers.

IP address

The IP address of a slave server in your domain.

3.2.2.3 Client

Setup options

The list of setup options for clients follows. Write the options you want to use in the appropriate place in the worksheet.

- Run the ypbind daemon with the –s option, which requires the server to use a reserved port.

  For security purposes, it is best to run NIS with the –s option.

- Lock the ypbind daemon to a particular domain name and server list by using the -S option.

Normally, hosts broadcast NIS requests on the network and the first available server answers the request. The –S option allows you to lock the ypbind daemon to a particular domain and set of servers. Requests are made directly to the specified servers, rather than being broadcast. For security purposes, it is best to run NIS with the –S option.

If you choose to run NIS with the –S option, you must know the host names and IP addresses of the servers to which you are locking the ypbind daemon to successfully complete the configuration process.

- Run NIS with the –ypset option or the –ypsetme option.

  The –ypset option allows a user logged in as root on any system in your domain to bind your system to a particular server. The –ypsetme option allows ypbind to accept –ypset requests only from the local system. For security purposes, it is best to disallow all ypset requests.

- Use Automount or AutoFS and the associated maps.

  The automount and autofsd daemons, which are alternatives to mounting remote file systems in the /etc/fstab file, allow users to mount remote file systems on an as-needed basis. When you use NIS to distribute the maps for these daemons, you can configure clients to receive the maps from the NIS master and slave servers and use the maps to mount remote file systems. For information on creating these maps, see Appendix A. For information on administering the maps, see Section 4.1.2.

  Whether or not you use Automount or AutoFS depends on your site's networking environment.

Server name

The name of a master or slave server in your domain. Specify several servers.

3.3 Configuring NIS

Use the SysMan Menu application of the Common Desktop Environment (CDE) Application Manager to configure NIS on master servers, slave servers, and clients. To invoke the SysMan Menu application, follow the instructions in Section 1.2.1.

3.3.1 Configuring an NIS Master Server

You must configure the NIS master server before you configure the other systems. Prior to using the SysMan Menu or the `nissetup` script, you must log in as root and complete the following tasks:

1. Copy into the `/var/yp/src` directory the local `/etc` files that you intend to make into NIS maps for distribution. If a file is absent from the `/var/yp/src` directory while it is building the default NIS maps, the `nissetup` script issues an informational message that it could not find that particular file and continues building the maps.

   Note

   If you copied the `passwd` file into the `/var/yp/src` directory, remove the `root` entry from the file.

2. Optionally, create the `/var/yp/src/mail.aliases` file. If you already have a `/var/adm/sendmail/aliases` file on your local system, you can copy it to the `/var/yp/src` directory and edit it, if necessary. For information on the format of this file, see `aliases`(4).
3. Optionally, create the `/var/yp/src/netgroup` file. For information on the format of this file, see `netgroup`(4).
4. Edit the `/var/yp/Makefile` file.

   If you are using the NIS master server to serve the `/etc/auto.master` and `/etc/auto.home` maps for Automount or AutoFS, you must remove the comment sign (#) from the beginning of each of the following lines. These lines were added to the `Makefile` for use by the `automount` and `autofsd` daemons.

```
   .
   .
#all: passwd group hosts networks rpc services protocols netgroup \
#     aliases auto.home auto.master
   .
   .
#$(YPDBDIR)/$(DOM)/auto.home.time: $(DIR)/auto.home
#        -@if [ -f $(DIR)/auto.home ]; then \
#               $(SED) -e "/^#/d" -e s/#.*$$// $(DIR)/auto.home | \
#               $(MAKEDBM) -a $(METHOD) - $(YPDBDIR)/$(DOM)/auto.home; \
#               $(TOUCH) $(YPDBDIR)/$(DOM)/auto.home.time; \
#               $(ECHO) "updated auto.home"; \
#               if [ ! $(NOPUSH) ]; then \
#                       $(YPPUSH) auto.home; \
#                       $(ECHO) "pushed auto.home"; \
#               else \
#                       : ; \
#               fi \
#       else \
#               $(ECHO) "couldn't find $(DIR)/auto.home"; \
#       fi
#
```

```
#$(YPDBDIR)/$(DOM)/auto.master.time: $(DIR)/auto.master
#        -@if [ -f $(DIR)/auto.master ]; then \
#             $(SED) -e "/^#/d" -e s/#.*$$// $(DIR)/auto.master | \
#              $(MAKEDBM) -a $(METHOD) - $(YPDBDIR)/$(DOM)/auto.master; \
#              $(TOUCH) $(YPDBDIR)/$(DOM)/auto.master.time; \
#              $(ECHO) "updated auto.master"; \
#              if [ ! $(NOPUSH) ]; then \
#                      $(YPPUSH) auto.master; \
#                      $(ECHO) "pushed auto.master"; \
#              else \
#                      : ; \
#              fi \
#       else \
#              $(ECHO) "couldn't find $(DIR)/auto.master"; \
#       fi
   .
   .
   .
#auto.home: $(YPDBDIR)/$(DOM)/auto.home.time
#auto.master: $(YPDBDIR)/$(DOM)/auto.master.time
   .
   .
   .
#$(DIR)/auto.home:
#$(DIR)/auto.master:
```

Place a comment sign (#) in front of the following lines:

```
all: passwd group hosts networks rpc services protocols netgroup \
aliases
```

If you are using the NIS master server to serve other site-specific maps, you must add entries for the maps to the `Makefile`. See Section 3.4.8.1 for information on adding entries for site-specific NIS maps, other than the `/etc/auto.master` and `/etc/auto.home` maps, to the `/var/yp/Makefile` file.

5. Copy the `auto.master` and `auto.home` maps, or any other site-specific maps, to the `/var/yp/src` directory. For information on creating Automount or AutoFS maps, see Appendix A. For information on creating other site-specific maps, see the Section 3.4.8.1.

To continue to set up the master server, invoke the SysMan Menu as documented in Section 1.2.1 and do the following:

1. From the SysMan Menu, select Networking→Additional Network Services→Configure Network Information Service (NIS). SysMan Menu invokes the `nissetup` script.

   Alternatively, enter the following command on a command line:

   ```
   # /usr/bin/sysman nis
   ```

   A message reminds you that your network must be established before setting up NIS, and that in order to set up an NIS server you must have the Additional Networking Services subset installed.

2. Enter `c` to continue.

3. Press Return following the script's explanation of nissetup, and then press Return again after the script explains the three types of systems in an NIS domain.
4. Enter and confirm your system's case-sensitive NIS domain name.
5. Choose option 1 to indicate that you are configuring the master server.
6. Following the nissetup script's explanation that there can be only one master server configured for each NIS domain, enter c and indicate whether or not you want to run the yppasswdd daemon. It is best to run the yppasswdd daemon on the NIS master server.
7. Indicate whether or not you intend to use enhanced security with NIS.
8. Indicate whether or not you want your NIS maps to be maintained as btree files.
9. Enter the names of hosts that will be slave servers for this domain. If you enter a host name that is not listed in the master server's /etc/hosts file, the nissetup script prompts you for its IP address.

```
Enter the names of the SLAVE servers in the test_domain domain.
Press Return to terminate the list.

   Host name of slave server: host2
   Host name of slave server: host3
      Cannot find host3 in the file /etc/hosts.
      To add host3 to the /etc/hosts file you MUST
                know host3's Internet (IP) address.

  Would you like to add host3 to the /etc/hosts file
        (y/n) [y]? y

  What is host3's Internet (IP) address [no default] ?
     120.105.1.28

  Is 120.105.1.28 correct (y/n) [no default] ? y

   Hostname of slave server:  Return
```

The nissetup script displays the list of servers that you entered. You can redo the list to correct errors or continue with the setup procedure.

The nissetup script then creates the default NIS maps, displaying messages similar to the following as it does:

```
Creating default NIS maps.  Please wait...
updated passwd
updated group
updated hosts
updated networks
updated rpc
updated services
updated protocols
updated netgroup
Finished creating default NIS maps.
```

10. Indicate whether or not you want to use the –s security option.

    If you choose to run NIS with the –s option, the ypbind process runs in a secure mode. It is best to use this option.

11. Indicate whether or not you want to use the –S security option.

    It is best to use this option. If you choose to run NIS with the –S option, you must enter the names of up to four NIS servers.

    If you enter the name of a server that is not listed in the system's /etc/hosts file, the nissetup script prompts you for its IP address. When you are done entering the list of servers, press Return on a blank Server *n* name field and enter c to continue configuring NIS on your system.

12. Indicate whether or not you want to allow ypset requests on your system.

    It is best to disallow all ypset requests. Press Return to accept the default, and confirm your choice.

13. Indicate whether or not you want your system to use all of the NIS databases served by the master server.

    It is best to use all of the NIS databases.

    If you choose to use all of the NIS databases, the nissetup script edits the /etc/svc.conf file to include the string yp for each database. It also edits the /etc/passwd and /etc/group files to include a plus sign followed by a colon (+:) at the end of each file. This enables your system to use NIS for each database listed. This symbol enables the files to be distributed by NIS. Continue with step 16.

    If you choose not to use all of the NIS databases, enter n and continue with the next step.

14. Indicate whether or not you want to add a plus sign followed by a colon (+:) to the end of the local /etc/passwd or /etc/group files.

    For your system to use the NIS-served passwd database, group database, or both, +: must be the last line in the file or files you want served by NIS. This applies to the passwd and group databases only.

Note

The service order selection for the passwd and group databases is handled by the Security Integration Architecture (SIA). If BSD is selected for passwd and group information in the /etc/sia/matrix.conf file, only the +: is required for your system to search NIS.

15. Indicate whether or not you want the `nissetup` script to invoke the `svcsetup` script.

    If you answer yes, the `nissetup` script invokes the `svcsetup` script, which allows you to modify the database services selection file (the `svc.conf` file). See Section 3.3.4 for information on modifying the `svc.conf` file.

    If you answer no, the `nissetup` script continues. You must edit the `svc.conf` file later if you want your system to use NIS to obtain database information other than `passwd` and `group` information.

16. Indicate whether or not to start the NIS daemons automatically.

    If you answer yes, `nissetup` starts the daemons.

    If you answer no, use the following command to start the daemons manually after `nissetup` exits and returns you to the system prompt (#):

    ```
    # /sbin/init.d/nis start
    ```

3.3.2 Configuring a Slave Server

To configure a slave server, invoke the SysMan Menu as documented in Section 1.2.1 and do the following:

1. From the SysMan Menu, select Networking→Additional Network Services→Configure Network Information Service (NIS). SysMan Menu invokes the `nissetup` script.

   Alternatively, enter the following command on a command line:

   ```
   # /usr/bin/sysman nis
   ```

2. A message reminds you that your network must be established before setting up NIS, and that in order to set up an NIS server you must have the Additional Networking Services subset installed. Enter `c` to continue.
3. Press Return following the script's explanation of `nissetup`, and then press Return again after the script explains the three types of systems in an NIS domain.
4. Enter and confirm your system's case-sensitive NIS domain name.
5. Choose option 2 to indicate that you are configuring a slave server.
6. Enter `c` to continue following the `nissetup` script's explanation that the master server's list must include each slave server, and that the master server must be established in order for maps to be copied to the slave server.
7. Enter the name of the master server for your domain.

8. Indicate whether or not you intend to use enhanced security with NIS.
9. Indicate whether or not you want your NIS maps to be maintained as btree files.

   After you indicate your choice, the script copies the default NIS maps from the master NIS server.
10. Indicate whether or not you want to use the –s security option.

    If you choose to run NIS with the –s option, the ypbind process runs in a secure mode. It is best to use this option.
11. Indicate whether or not you want to use the –S security option.

    It is best to use this option. If you choose to run NIS with the –S option, you must enter the names of up to four NIS servers.

    If you enter the name of a server that is not listed in the system's /etc/hosts file, the nissetup script prompts you for its IP address. When you are done entering the list of servers, press Return in a blank Server *n* name field and enter c to continue configuring NIS on your system.
12. Indicate whether or not you want to allow ypset requests on your system.

    It is best to disallow all ypset requests. Press Return to accept the default and confirm your choice.
13. Indicate whether or not you want your system to use all of the NIS databases served by the master server.

    It is best to use all of the NIS databases.

    If you choose to use all of the NIS databases, the nissetup script edits the /etc/svc.conf file to include the string yp for each database. It also edits the /etc/passwd and /etc/group files to include a plus sign followed by a colon (+:) at the end of each file. This enables your system to use NIS for each database listed. This symbol enables the file to be distributed by NIS. Continue with step 16.

    If you choose not to use all of the NIS databases, enter n and continue with the next step.
14. Indicate whether or not you want to add +: to the end of the local /etc/passwd or /etc/group files.

    For your system to use the NIS-served passwd database, group database, or both, +: must be the last line in the file or files you want NIS to serve. This applies to the passwd and group databases only.

Note

The service order selection for the `passwd` and `group` databases is handled by the Security Integration Architecture (SIA). If `BSD` is selected for `passwd` and `group` information in the `/etc/sia/matrix.conf` file, the `+:` is required only for your system to search NIS.

15. Indicate whether or not you want the `nissetup` script to invoke the `svcsetup` script.

    If you answer yes, the `nissetup` script invokes the `svcsetup` script, which allows you to modify the database services selection file (the `svc.conf` file). See Section 3.3.4 for information on modifying the `svc.conf` file.

    If you answer no, the `nissetup` script continues. You must edit the `svc.conf` file later if you want your system to use NIS to obtain database information other than `passwd` and `group` information.

16. Indicate whether or not to start the NIS daemons automatically.

    If you answer yes, `nissetup` starts the daemons.

    If you answer no, use the following command to start the daemons manually after `nissetup` exits and returns you to the system prompt (#):

    ```
    # /sbin/init.d/nis start
    ```

3.3.3 Configuring an NIS Client

To configure an NIS client, invoke the SysMan Menu as documented in Section 1.2.1 and do the following:

1. From the SysMan Menu, select Networking→Additional Network Services→Configure Network Information Service (NIS). SysMan Menu invokes the `nissetup` script.

   Alternatively, enter the following command on a command line:

   ```
   # /usr/bin/sysman nis
   ```

2. A message reminds you that your network must be established before setting up NIS, and that in order to set up an NIS server you must have the Additional Networking Services subset installed. Enter `c` to continue.

3. Press Return following the script's explanation of `nissetup`, and then press Return again after the script explains the three types of systems in an NIS domain.

4. Enter and confirm your system's case-sensitive NIS domain name.

5. Press Return to accept the default that you are configuring a client.
6. Enter `c` to continue following the `nissetup` script's warning that at least one server must be configured for this domain.
7. Indicate whether or not you want to use the `–s` security option.

   If you choose to run NIS with the `–s` option, the `ypbind` process runs in a secure mode. It is best to use this option.
8. Indicate whether or not you want to use the `–S` security option.

   It is best to use this option. If you choose to run NIS with the `–S` option, you must enter the names of up to four NIS servers.

   If you enter the name of a server that is not listed in the system's `/etc/hosts` file, the `nissetup` script prompts you for its IP address. When you are done entering the list of servers, press Return in a blank `Server` *`n`* `name` field and enter `c` to continue configuring NIS on your system.
9. Indicate whether or not you want to allow `ypset` requests on your system.

   It is best to disallow all `ypset` requests. Press Return to accept the default, and confirm your choice.
10. Indicate whether or not you want your system to use all of the NIS databases served by the master server.

    It is best to use all of the NIS databases.

    If you choose to use all of the NIS databases, the `nissetup` script edits the `/etc/svc.conf` file to include the string `yp` for each database. It also edits the `/etc/passwd` and `/etc/group` files to include a plus sign followed by a colon (`+:`) at the end of each file. This enables your system to use NIS for each database listed. This symbol enables the file to be distributed by NIS. Continue with step 13.

    If you choose not to use all of the NIS databases, enter `n` and continue with the next step.
11. Indicate whether or not you want to add `+:` to the end of the local `/etc/passwd` or `/etc/group` files.

    For your system to use the NIS served `passwd` database, `group` database, or both, `+:` must be the last line in the file or files you want served by NIS. This applies to the `passwd` and `group` databases only.

Note

The service order selection for the `passwd` and `group` databases is handled by the Security Integration Architecture (SIA). If `BSD` is selected for password and group information

in the /etc/sia/matrix.conf file, the +: is required only for your system to search NIS.

---

12. Indicate whether or not you want the nissetup script to invoke the svcsetup script.

    If you answer yes, the nissetup script invokes the svcsetup script, which allows you to modify the database services selection file (the svc.conf file). See Section 3.3.4 for information on modifying the svc.conf file.

    If you answer no, the nissetup script continues. You must edit the svc.conf file later if you want your system to use NIS to distribute database information other than password and group information.

13. Indicate whether or not to start the NIS daemons automatically.

    If you answer yes, nissetup starts the daemons.

    If you answer no, use the following command to start the daemon manually after nissetup exits and returns you to the system prompt (#):

    ```
    # /sbin/init.d/nis start
    ```

3.3.4 Modifying the svc.conf File with svcsetup

If you choose not to use NIS for all of the default databases, you can edit the /etc/svc.conf file with the svcsetup script. If you answer yes when nissetup asks if you want to run svcsetup, it invokes the svcsetup script. Use the following procedure to edit the /etc/svc.conf file:

1. Press Return to choose the m option from the Configuration Menu.
2. Enter the numbers from the Change Menu that correspond to the databases whose entries you want to modify.
3. Enter the number that corresponds to the order in which you want to query the services on your system.

   If you choose the default (2), the local /etc files are searched first for the requested information. If the information is not found locally, then an NIS server is queried. This choice is valid for all of the databases that NIS serves.

   To have NIS serve hosts information if your system is also having hosts information served by DNS, choose either option 5 (local,bind,yp) or option 6 (bind,local,yp) for the hosts database. Note that options 3 (local,bind), 4 (bind,local), 5, and 6 are valid for the hosts database only.

3.3.5 Modifying or Removing an NIS Configuration

If you configure NIS and run the `nissetup` script, you can modify or remove the NIS configuration.

If you choose to modify the NIS configuration, the `nissetup` script proceeds as described in Section 3.3.1 to Section 3.3.3, resulting in a new configuration.

If you choose to remove the NIS configuration, the `nissetup` script prompts you to verify your choice, then removes the NIS information from the following files:

- `/etc/rc.config.common`
- `/etc/passwd`
- `/etc/group`
- `/etc/svc.conf`
- `/var/yp/`*`DOMAIN`* (where *`DOMAIN`* is the name of the current NIS domain)

  This directory and its contents are deleted (for NIS master and slave servers only).

3.4 Managing an NIS Server

This section describes how to perform the following NIS server tasks:

- Add an NIS slave server to a domain
- Remove an NIS slave server from a domain
- Add a user to an NIS domain
- Update an NIS map
- Add an NIS map to a domain
- Remove an NIS map from a domain
- Modify the `/var/yp/Makefile` file
- Restrict access to NIS data

3.4.1 Adding an NIS Slave Server to a Domain

Adding a slave server to a domain enables the slave server to receive updated NIS maps from the master server and distribute them to NIS clients in a domain.

To add an NIS slave server to a domain, do the following:

1. Set up the system as a slave server. See Section 3.3.2 for information on setting up a slave server.
2. Log in to the NIS master server as root.
3. Change to the `/var/yp` directory by using the `cd` command.
4. Undo the `ypservers` map and direct the output to a file by executing the following command:

   ```
   # ./makedbm -a method -u domainname/ypservers > filename
   ```

   For *method*, specify the letter for the appropriate database format. The letter is `d` for dbm/ndbm (the default database format), `b` for btree, or `h` for hash.

   Specify your NIS domain name for *domainname* and a temporary file name for *filename*.
5. Edit the output file from the `makedbm` operation and add the host name of the slave server.
6. Build a new `ypservers` map and replace the old one by executing the following command:

   ```
   # ./makedbm -a method filename domainname/ypservers
   ```

   Again, for *method*, specify the letter for the appropriate database format. Specify the NIS domain name for *domainname* and the temporary file name for *filename*.

   You can optionally combine steps 4, 5, and 6 into one command line. See the example at the end of this procedure.
7. Distribute the updated `ypservers` map to the slave servers by using the `yppush` command.

   ```
   # ./yppush ypservers
   ```
8. Edit the NIS master server's `hosts` source file and add an entry for the slave server, if it is not already in the `hosts` file. Then, update the map by entering the `make` command. The `make` command also distributes the updated map:

   ```
   # make hosts
   ```

See `makedbm`(8) for more information on building maps.

The following example (illustrating steps 3 through 8) shows how to add slave server `host8` to domain `market`. It assumes that the maps are in `btree` format:

```
# cd /var/yp
# (./makedbm -a b -u market/ypservers ; echo host8 host8)\ 1
```

```
 |./makedbm -a b - market/ypservers
# ./yppush ypservers [2]
# vi ./src/hosts [3]
   .
   .
   .
# make hosts [4]
```

[1] Represents the combination of steps 4, 5, and 6 in the preceding procedure. The output from the `makedbm` command is displayed and the new server name, `host8`, is echoed on standard output to add it to the file. Then, the output is piped back into the `makedbm` command to build the new map, which overwrites the old `ypservers` map.

Note

You can type these lines as one command even if the command wraps on your screen, or you can use the backslash escape character (\), as shown.

[2] Distributes the updated map to the slave servers.

[3] Adds the new slave server to the `hosts` NIS source file on the master server, if necessary.

[4] Updates the map and distributes the updated `hosts` map to the slave servers.

Section B.1 contains a sample script you can copy that performs the steps involved in adding a slave server to a domain. You still have to set up the slave server and edit the master server's `hosts` file, adding a slave server entry, if necessary.

3.4.2 Removing an NIS Slave Server from the Domain

Removing a slave server from a domain means that the system will no longer receives updated NIS maps from the master server and distribute them to NIS clients in a domain.

To remove an NIS slave server from the domain, do the following:

1. Log in to the NIS slave server.

   If the system will be an NIS client, configure it as an NIS client by using `nissetup`. See Section 3.3.3 for more information.

   If the system will no longer use NIS, disable NIS in the `/etc/rc.config.common` file by using the following command:

   ```
   # /usr/sbin/rcmgr –c set NIS_CONF NO
   ```

2. Log in to the NIS master server as root.

3. Change to the /var/yp directory by using the cd command.
4. Undo the ypservers map and direct the output to a file by executing the following command:

   # **./makedbm -a *method* -u *domainname*/ypservers > *filename***

   For *method*, specify the letter for the appropriate database format. The letter is d for dbm/ndbm (the default database format), b for btree, or h for hash.

   Specify your NIS domain name for *domainname* and a temporary file name for *filename*.
5. Edit the output file from the makedbm operation and remove the host name of the slave server.
6. Build a new ypservers map by executing the following command:

   # **./makedbm -a *method filename domainname*/ypservers**

   Again, for *method*, specify the letter for the appropriate database format. Specify the NIS domain name for *domainname* and the temporary file name for *filename*.

   You can optionally combine steps 4, 5, and 6 into one command line. See the example at the end of this procedure.
7. Distribute the updated ypservers map to the slave servers by using the yppush command:

   # **./yppush ypservers**
8. If necessary, edit the NIS master server's hosts source file and remove the entry for the slave server. Then, update the hosts map by entering the make command. The make command also distributes the updated map:

   # **make hosts**

See makedbm(8) for more information on building maps.

The following example (illustrating steps 3 through 8) shows how to remove slave server host4 from domain market. It assumes that the maps are in btree format:

```
# cd /var/yp
# ./makedbm -a b -u market/ypservers |\ 1
 grep –v host4 | ./makedbm -a b - market/ypservers
# ./yppush ypservers 2
# vi ./src/hosts 3
   ⋮
```

```
# make hosts 4
```

1 Represents the combination of steps 4, 5, and 6 in the preceding procedure. The output from the `makedbm` command is piped into `grep` with the –v option to display all lines except the one containing the slave server name (`host4`). Then, the output is piped back into the `makedbm` command to build the new map, which overwrites the old `ypservers` map.

Note

You can type these lines as one command even if the command wraps on your screen, or you can use the backslash escape character (\), as shown.

2 Distributes the updated map to the slave servers.

3 Removes the slave server from the `hosts` NIS source file on the master server, if necessary.

4 Updates the `hosts` map and distributes the updated map to the slave servers.

Section B.2 contains a sample script you can copy that performs the steps involved in removing a slave server from a domain. You still have to reconfigure the slave server as an NIS client or as a system that does not use NIS.

3.4.3 Adding a New User to an NIS Domain

Adding a new user to an NIS domain adds the user's account information to the `passwd` map and allows the user to participate in the NIS environment. A user has only one password on all systems that use NIS for their `passwd` map.

To add a new user to an NIS domain, invoke the SysMan Menu on the NIS master server, as documented in Section 1.2.1, and do the following:

1. From the SysMan Menu, select Accounts→Manage NIS Users to display the Manage NIS Users dialog box.

   Alternatively, enter the following command on a command line:

   ```
   # /usr/bin/sysman nis_users
   ```

2. Select Add to display the Add a User dialog box.
3. Enter the user name, user ID, and password for the new user.

4. Select a primary group for the user:

   a. Select Choose to open the Primary Group dialog box.

   b. Select one group from the list of groups. Then, select OK to close the Primary Group dialog box.

5. Enter a secondary group for the user, if necessary.

6. Select a shell for the user.

   a. Select Choose to open the Shells dialog box.

   b. Select a shell from the pull-down menu. Then, select OK to close the Primary Group dialog box.

7. Deselect the Create Home Directory check box if you do not want the system to create a home directory for the user. By default, the system creates a directory for the user in the `/usr/users` directory.

   If you choose to allow the system to create the user's home directory, you can specify an alternate location for the directory in the Home Directory field.

8. Enter comments for the account, if necessary. For example, at a college, you could use this field to indicate that a new account is temporary for a visiting professor.

9. Deselect the Lock Account check box to unlock the account. Unlocking the account gives the user permission to log in and use the account.

10. Select OK to create the user's account. You are informed that the account has been created. Select OK to dismiss the confirmation message and to close the Add a User dialog box.

11. Select Exit to close the Manage NIS Users dialog box.

12. Create the user's home directory if you did not allow the utility to create it for you. Then, set up the user's environment. See the *System Administration* manual for more information.

You can also modify and delete NIS user accounts with the SysMan Menu. See the online help for more information.

If you prefer, you can use the `dxaccounts` or `useradd` utilities to administer NIS users. See the online help and `useradd`(8) for more information.

3.4.4 Adding a New Group to an NIS Domain

Adding a group to an NIS domain adds the group and all of its registered users to the `group` map. To add a new group to an NIS domain, invoke the

SysMan Menu on the NIS master server, as documented in Section 1.2.1, and do the following:

1. From the SysMan Menu, select Accounts→Manage NIS Groups to display the Manage NIS Groups dialog box.

   Alternatively, enter the following command on a command line:

   ```
   # /usr/bin/sysman nis_groups
   ```

2. Select Add to display the Add a Group dialog box.
3. Enter the group name and group ID for the new group.
4. Select one or more users who will be in the group from the Members list.
5. Select OK to create the group. You are informed that the group has been created. Select OK to dismiss the confirmation message and to close the Add a Group dialog box.
6. Select Exit to close the Manage NIS Groups dialog box.

You can also modify and delete NIS groups with the SysMan Menu. See the online help for more information.

If you prefer, you can use the `dxaccounts` or `groupadd` utilities to administer NIS groups. See the online help and `groupadd`(8) for more information.

3.4.5 Updating an NIS Map

Updating an NIS map involves making changes to an NIS map's master file, updating the `Makefile` file (if the map is not listed), and building and distributing the new map. Entries for the following standard maps are included in the `Makefile` file:

- `passwd`
- `group`
- `hosts`
- `networks`
- `rpc`
- `services`
- `protocols`
- `netgroup`
- `aliases` (`mail.aliases`)

The master files are located in the `/var/yp/src` directory on the NIS master server.

To update an NIS map, do the following:

1. Log in to the NIS master server as root.
2. Change to the `/var/yp` directory by using the `cd` command.
3. Modify the `Makefile` file, if no entry exists in the `/var/yp/Makefile` file for the map you want to update.

   See Section 3.4.8 for information on modifying the `Makefile` file.
4. Change to the `/var/yp/src` directory by using the `cd` command.
5. Edit the master file of the map you want to update and make your changes.
6. Change to the `/var/yp` directory by using the `cd` command.
7. Update and distribute the map by using the `make` command as follows:

   ```
   # make map_name
   ```

The following example (illustrating steps 4 through 7) shows how to update the `hosts` map:

```
# cd var/yp/src [1]
# vi hosts [2]
   ⋮
# cd /var/yp [3]
# make hosts [4]
```

[1] Changes to the `/var/yp/src` directory.

[2] Opens the `/var/yp/src/hosts` file for editing.

[3] Changes to the `/var/yp` directory.

[4] Updates the map and distributes it to the slave servers.

3.4.6 Adding an NIS Map to a Domain

Adding an NIS map to a domain allows the database information to be distributed throughout an NIS domain. You can create and distribute maps for any information you want to distribute.

To add an NIS map to a domain, do the following:

1. Log in to the NIS master server as root.
2. Create a master file for your new map.

   A master file is an ASCII text file containing individual entries. Each entry has fields separated by spaces. Some of these fields are used to build a key to each entry. Review some of the master files in the

/var/yp/src directory to better understand the structure of a master file.

3. If you are using NIS to distribute NFS Automount or AutoFS maps, create a file named auto.master in the /var/yp/src directory. If the file exists, add an entry for the map you want to distribute.

   See Section 4.1.2 and Appendix A for more information on the auto.master map.

4. Edit /var/yp/Makefile file to include the new map in the default set of maps.

   See Section 3.4.8 for information on modifying the Makefile file.

5. Change to the /var/yp directory by using the cd command.

6. Update the map by using the make command as follows:

   # **make *map_name***

The following example adds the phonelist map to a domain:

```
# vi /var/yp/src/phonelist [1]
   .
   .
   .
# vi /var/yp/Makefile [2]
   .
   .
   .
# cd /var/yp [3]
# make phonelist [4]
```

[1] Creates a phonelist master file on the master server.

[2] Opens the Makefile file for editing.

[3] Changes to the /var/yp directory.

[4] Updates the map and distributes the updated map to the slave servers.

3.4.7 Removing an NIS Map from a Domain

Removing an NIS map from a domain prevents the database information from being distributed throughout an NIS domain.

To remove an NIS map from a domain, do the following:

1. Log in to the NIS master server as root.

2. If you are using NIS to distribute NFS Automount or AutoFS maps, delete the entry for the map you no longer want distributed from the auto.master file in the /var/yp/src directory.

   See Section 4.1.2 and Appendix A for more information on the auto.master map.

3. Edit the /var/yp/Makefile file to remove the map from the default set of maps.

   See Section 3.4.8 for information on modifying the Makefile file.

3.4.8 Modifying the /var/yp/Makefile File

Modifying the Makefile file means adding or deleting database entries in the /var/yp/Makefile file on the NIS master server. By adding a database entry to the Makefile file, you indicate that you want a map produced for the specific database when you use the make command. By deleting a database entry, you indicate that you do not want a map produced for the specific database.

As you edit the /var/yp/Makefile file, remember the following:

- The order of entries in the line that begins with all: is not important. However, in continuation lines, the blank space preceding the line must be a tab character; do not use spaces.
- Variables are defined at the top of the Makefile file.

3.4.8.1 Adding an Entry

To add an entry to the Makefile file, do the following:

1. Log in to the NIS master server as root.
2. Edit the /var/yp/Makefile file and add the database name to the line beginning with all:. Next, add a line with the following format to the end of the file:

   database_name:*database_name*.time

   Finally, add an entry with the following format to the middle of the file:

   database_name.time: *various_commands*

   To simplify the creation of this entry, copy the auto.home.time: entry in the file and make the necessary database name changes.
3. If you are using NIS to distribute NFS Automount or AutoFS maps, uncomment any line that contains the auto.master string by deleting the comment character (#) that precedes it.

The following example shows the phonelist database added to the /var/yp/Makefile file. There is a tab character preceding the netgroup database name in the all: line.

```
all: passwd group hosts networks rpc services protocols \
        netgroup aliases phonelist
   ⋮
$(YPDBDIR)/$(DOM)/phonelist.time: $(DIR)/phonelist
```

```
        -@if [-f $(DIR)/phonelist ]; then \
             $(SED) -e "/^#/d" -e s/#.*$$// $(DIR)/phonelist | \
             $(MAKEDBM) -a $(METHOD) - $(YPDBDIR)/$(DOM)/phonelist; \
             $(TOUCH) $(YPDBDIR)/$(DOM)/phonelist.time; \
             $(ECHO) "updated phonelist"; \
             if [ ! $(NOPUSH) ]; then \
                     $(YPPUSH) phonelist; \
                     $(ECHO) "pushed phonelist"; \
             else \
                     : ; \
             fi \
 else \
             $(ECHO) "couldn't find $(DIR)/phonelist"; \
 fi    ⋮
phonelist: phonelist.time
```

3.4.8.2 Deleting an Entry

To delete an entry from the `Makefile` file, do the following:

1. Log in to the NIS master server as root.
2. Edit the `/var/yp/Makefile` file, delete the database name from the line beginning with `all:`, and delete the line beginning with the database name (`database_name:`).

   Instead of deleting the database line, you can comment out the line by adding a comment character (#) to the beginning of the line.

3.4.9 Restricting Access to NIS Data

By default, the `ypserv` and `ypxfrd` daemons provide NIS information to anyone with network access to an NIS server who makes a request. However, you can restrict NIS database access to only those hosts in subnets you specify by completing the following steps:

1. Log in to the NIS server as root.
2. Create a `/var/yp/securenets` file.
3. Edit the `/var/yp/securenets` file and add an entry for each subnet from which the NIS server is to accept NIS requests. The format of each file entry is as follows:

   subnet_mask subnet_ip_address

   For example:

   ```
   255.255.0.0  128.30.0.0 [1]
   255.255.255.0 128.211.10.0 [2]
   ```

```
255.255.255.255 128.211.5.6 [3]
```

[1] Allows IP addresses that are within the subnet `128.30` range to access the NIS files. The network mask is `255.255.0.0` and the corresponding network address is `128.30.0.0`.

[2] Allows IP addresses that are within the subnet `128.211.10` range to access the NIS files.

[3] Allows one host with the IP address `128.211.5.6` to access the NIS files.

4. Save the file.

If the file does not exist or contains no entries, the server accepts any NIS request.

If the file exists and contains entries, the `ypserv` and `ypxfrd` daemons read the `/var/yp/securenets` file during initialization. When an NIS request is received, the requester's IP address is compared to the subnets in the `/var/yp/securenets` file. If it matches, the request is processed. If it does not match, NIS silently discards the request. No message is logged (because malicious users could use these messages to fill up a system's disk).

On the system making the NIS request, NIS commands such as `ypcat` terminate with no error message. If a user is trying to log in to a system, the login times out after many retries.

Note

If the `/var/yp/securenets` file is modified, you must kill and restart the `ypserv` and `ypxfrd` daemons.

You can also use a `/var/yp/securenets` file to restrict access to NIS data on a slave server. However, the NIS slave server's IP address must be in the authorization range of entries in the `/var/yp/securenets` file of the NIS master.

3.5 Managing an NIS Client

This section describes how to perform the following NIS client management tasks:

- Change an NIS password
- Obtain NIS map information

3.5.1 Changing an NIS Password

To change a user's password in the NIS `passwd` map, use the `yppasswd` command. If you receive an error message, ask the system administrator on the master server to verify that the `rpc.yppasswdd` daemon on the NIS master server is running.

If you try to change an NIS-distributed password with the `passwd` command, you receive the following error message:

```
Not in passwd file.
```

The root password is local and not in the NIS file. To change the root password, use the `passwd` command.

See `yppasswd`(1) and `rpc.yppasswdd`(8) for further information.

3.5.2 Obtaining NIS Map Information

NIS map information includes the following:

- Map names
- Map values
- Map keys
- Map master server

To obtain NIS map information, issue one of the commands listed in Table 3–1.

Table 3–1: NIS Map Information Commands

Command	Action
`ypcat`	Prints values from an NIS database
`ypwhich`	Prints the name of the host that is the current NIS server or map master
`ypmatch`	Prints the values of one or more keys from an NIS map

Use the `–x` option with any of the commands shown in Table 3–1 to list all the map nicknames.

See `ypcat`(1), `ypwhich`(1), and `ypmatch`(1) for more information about these commands.

The following command lists all available maps and their master servers:

```
# ypwhich –m
```

The following command lists all values in the `hosts` map:

```
# ypcat hosts
```

The following command lists all occurrences in the `hosts` map that have the key `apple`:

```
# ypmatch apple hosts
```

The following command lists all occurrences in the `hosts` map that have the name `jones` associated with them. The name `jones` is not a key in this map.

```
# ypcat hosts | grep jones
```

4

Network File System

The Network File System (NFS) is a facility for sharing files in a heterogeneous environment. This chapter describes:

- The NFS environment (Section 4.1)
- How to plan for your NFS configuration (Section 4.2)
- How to configure NFS servers and clients (Section 4.3)
- How to deconfigure NFS servers and clients (Section 4.4)
- How to manage an NFS server (Section 4.5)
- How to manage an NFS client (Section 4.6)

For introductory information on NFS, see `nfs_intro`(7). For troubleshooting information, see Section 9.8 for clients and Section 9.7 for servers.

4.1 NFS Environment

In the NFS environment, systems can have the following roles:

- Client — A system that imports file systems. A client can mount file systems by using either the `/etc/fstab` file or an automatic mounting daemon, such as the `automount` or `autofsd` daemons. All methods are explained in this chapter.
- Server — A system that exports file systems.

Your system can be set up as an NFS server, a WebNFS server, an NFS client, or all three.

4.1.1 Distributing the hosts Database

If your network is running the Network Information Service (NIS) or the Domain Name System (DNS) to distribute host information, you do not need to list each server that is referenced in a client's `/etc/fstab` file in the client's local `/etc/hosts` file. However, the server's host information must be in the NIS or DNS database.

Similarly, if your network is running NIS or DNS to distribute host information and the client information is listed in the `hosts` database, you do not have to list each client that is referenced in a server's `/etc/exports` file in the server's local `/etc/hosts` file.

4.1.2 Automatic Mounting Daemons

The `automount` and `autofsd` daemons offer alternatives to mounting remote file systems with the `/etc/fstab` file, allowing you to mount them on an as-needed basis.

When a user on a system running one of these daemons invokes a command that must access a remotely mounted file or directory, the daemon mounts that file system or directory and keeps it mounted for as long as the user needs it. When a specified amount of time elapses (the default is 5 minutes) without the file system or directory being accessed, the daemon unmounts it.

You specify the file systems to be mounted in map files. These maps can be customized to suit your environment and are administered in the following ways:

- Use NIS to create and distribute the maps
- Administer the maps locally
- Use a combination of both methods

See Appendix A for information on creating these maps. With a few restrictions, as documented in the Restrictions section of `autofsmount`(8), Automount and AutoFS maps can be used interchangeably.

Note

The Automount daemon will be retired in a future release of the operating system. For information about migrating from Automount to AutoFS, see Section 4.6.3.5.

4.1.2.1 Serving Automount and AutoFS Maps with NIS

NIS allows you to create and distribute customized Automount and AutoFS maps. When NIS is used to distribute maps, the administrator of the NIS master server creates and administers the maps for the NIS domain. In this case, you must configure each system that uses Automount or AutoFS as an NIS client so that it can receive the maps.

If many clients in an environment remotely mount the same file system by specifying it in their `/etc/fstab` file, that file system is a good candidate for inclusion in a map distributed by NIS. Carefully constructed maps can allow client systems to eliminate a large part of their `/etc/fstab` files. If the location of a file system that is included in a distributed map changes, or its server changes, the administrator changes the map on the NIS master server. The change is then propagated throughout the domain without users on the client systems having to edit their `/etc/fstab` files.

See Section 3.3.1 for information on configuring a master NIS server to serve maps.

4.1.2.2 Local Automount and AutoFS Maps

Local Automount and AutoFS maps might be useful to you under the following circumstances:

- Your system mounts remote file systems that are not typically mounted by other NIS clients.
- Your network is not running NIS.
- You need to test a map.

Administering the `automount` or `autofsd` daemons locally is the same as administering them when NIS distributes the maps, except that you, as administrator of your system, create and manage the maps.

A local `auto.master` map serves the same function as one distributed in an NIS domain. If you specify a local `auto.master` map, the daemon consults it for the location of other maps, their local mount points, and the mount options. You can use an `auto.master` map that is distributed by NIS, a local `auto.master` map, both, or neither, if the selected daemon is invoked correctly.

4.1.2.3 WebNFS

WebNFS is an NFS protocol that allows clients to access files over the Internet in the same way that local files are accessed. WebNFS uses a public file handle that allows it to work across a firewall. This public file handle also reduces the amount of time required to initialize a connection. The public file handle is associated with a single directory (`public`) on the WebNFS server. See `exports`(4), `exportfs`(2), and `nfs_intro`(4) for further information.

4.2 Planning NFS

Figure 4–1 shows the NFS Setup Worksheet, which you can use to record the information required to configure NFS. If you are viewing this manual online, you can use the print feature to print a copy of this worksheet. The following sections explain the information you need to record on the worksheet.

Figure 4–1: NFS Setup Worksheet

NFS Setup Worksheet

Server

Number of nfsd threads: TCP: _____ UDP: _____

Property lists: ☐ Yes ☐ No

NFS locking: ☐ Yes ☐ No

PC-NFS daemon: ☐ Yes ☐ No

Allow nonroot mounts: ☐ Yes ☐ No

Address verification: ☐ Yes ☐ No

Exported directories:

Path name:	Permissions:	Network group/ Node name:

Client

Number of I/O threads: _____

NFS locking: ☐ Yes ☐ No

Automatic mounting daemon: ☐ Automount ☐ AutoFS ☐ None

Imported directories:

Path name:	Remote server name:	Local mount point:	RO:
			☐
			☐
			☐
			☐
			☐

4.2.1 Server

Number of nfsd threads

Enter the number of `nfsd` TCP and UDP server threads to run. These threads service requests from NFS clients. The default number of 8 is adequate for an average work load. You can configure a combined total of 0 to 128 TCP and UDP server threads.

On systems that support Cache Coherent NUMA, the number of threads is per Resource Affinity Domain (RAD). See `nfsd`(8) and `numa_intro`(3) for more information.

Property lists

If you want to run the property list daemon, check Yes; otherwise, check No. The property list daemon allows the server to handle requests to get, set, or delete the property lists associated with NFS-served file system objects. See `proplistd`(8) and `proplist`(4) for more information.

NFS locking

If you want to run the NFS lock manager (`rpc.lockd`) and status monitor (`rpc.statd`), check Yes. Running these daemons allows users to use the `fcntl` and `lockf` functions to lock file regions on NFS files (in addition to local files). If you do not run these daemons, users can use advisory locking primitives only on local files. For more information on the `fcntl` and `lockf` functions, see `fcntl`(2) and `lockf`(3).

PC-NFS daemon

If you want to run the PC-NFS daemon (`rpc.pcnfsd`), check Yes; otherwise, check No. The PC-NFS daemon allows the server to handle NFS requests from PCs.

Allow nonroot mounts

If you allow nonroot mounts, users on client systems who do not have root privileges can still mount the file systems or directories exported from this system. If you do not allow nonroot mounts, only the superusers on the client systems can mount file systems from this host. The default setting does not allow nonroot mounts.

Address Verification

If you want the server to verify the Internet address of any host that requests an exported directory, check Yes; otherwise, check No. If you choose Yes and you also want to verify that the host is in the server's domain or subdomain, check Domain Checking, Subdomain Checking, or both.

4.2.1.1 Exported Directories

Use the following fields to define file systems that your server will export to client systems:

Path name

The path name of the file systems or directories that you intend to export.

Permissions

The permissions to assign for each exported file system or directory. You can specify whether a file system or directory is exported with read-write (rw) or read-only (ro) permission, and you can map client superuser access to a root user ID (UID) number other than the default of –2. If you have a WebNFS server with the `-public` option set, the mount access list is ignored by the server so that all hosts using the WebNFS protocol have access to this directory. For more information on assigning permissions to exported file systems or directories and on specifically mapping the root UID for clients, see `exports`(4).

Network group/Node name

The network groups or individual host names to which you will export these file systems or directories. For information on defining network groups, see `netgroup`(4).

If you want to limit the hosts that can import a file system or directory, you must explicitly specify the individual hosts or network groups in the `/etc/exports` file. If you do not specify individual hosts or network groups, all hosts can import that file system or directory.

If you are exporting a file system to a client that has multiple network interfaces on a subnet, you must specify the host names for all of the interfaces; otherwise, export requests from the unspecified interfaces will be denied. See *Network Administration: Connections* for more information about multiple interfaces in a subnet and connection balancing.

4.2.2 Client

Number of I/O threads

The number of I/O threads to run. The default number of 7 is recommended for optimum load generation on servers. You can configure from 0 to 64 `nfsiod` threads.

In addition, you can start `nfsiod` threads from the command line. See `nfsiod`(8) for information on starting `nfsiod` threads from the command line.

NFS locking

If you want to run the NFS lock manager (`rpc.lockd`) and status monitor (`rpc.statd`), check Yes. Running these daemons allows users to use the `fcntl` and `lockf` functions to lock file regions on NFS files (in addition to local files). If you do not run these daemons, users can

use advisory locking primitives only on local files. For more information on the `fcntl` and `lockf` functions, see `fcntl`(2) and `lockf`(3).

Automatic mounting daemon

If the client is to run an automatic mounting daemon, such as Automount or AutoFS, check the box for one of these daemons; otherwise, check None.

You can select only one automatic mounting daemon. While AutoFS provides higher efficiency and availability than Automount, there are some restrictions for its use. See the Restrictions sections of `autofsd`(8) and `autofsmount`(8) for more information.

If the network is running the NIS, the Automount or AutoFS maps are better administered and served from the NIS master server. The format of the maps is the same whether they are local or served by the NIS master server. For information on creating maps, see Appendix A.

4.2.2.1 Imported Directories

Use the following fields to define the remote file systems that your client will import:

Path name

The complete pathnames of the file systems or directories that you want to import.

Remote server name

The host names of the servers from which you are importing file systems or directories.

Local mount point

The mount point on the local system where you want the imported file systems or directories to reside.

RO (Read-only)

The permissions for the imported file systems or directories. Check the box for a read-only mount. Leave the box unchecked for a read-write mount.

Note

If you mount your user area from a server, make sure that your UID on the client is the same as your UID on the server. NFS uses your client UID to check against file

access permissions on the server. If your UID is different on the client and server, you cannot modify your own NFS mounted files (assuming that you have the permissions on the mounted files set so that only you can modify them). Since the server does the access checking, the only UID allowed to modify the files is the one that the server knows.

4.3 Configuring NFS

Use the SysMan Menu application of the Common Desktop Environment (CDE) Application Manager to configure NFS on clients and servers. To invoke the SysMan Menu application, follow the instructions in Section 1.2.1.

4.3.1 Configuring an NFS Server

To configure an NFS server, complete the following steps. If you want your system to import file systems, see Section 4.3.2 for information on configuring an NFS client.

1. From the SysMan Menu, select Networking→Additional Network Services→Network File System (NFS)→Configure system as an NFS server to display the Configure NFS Server dialog box.

   Alternatively, enter the following command on a command line:

   ```
   # /usr/bin/sysman nfs_server
   ```

2. Enter the number of server TCP threads to be run in the appropriate field.
3. Enter the number of server UDP threads to be run in the appropriate field.
4. Select the Enable Property List Daemon check box if you want to run the property list daemon (`proplistd`).
5. Deselect the Enable Locking check box if you do not want to run the NFS lock manager (`rpc.lockd`) and status monitor (`rpc.statd`) daemons. Locking is enabled by default.
6. Select the Enable PC-NFS Daemon check button if you want to run the `rpc.pcnfsd` daemon.

   If you run the PC-NFS daemon, you must export to the client the directories you want to mount on the PC client. To enable the client to utilize network printing, you must export the `/usr/spool/pcnfs` directory to the PC client. For information on exporting directories, see Section 4.5.2.

7. Select the Allow Nonroot Mounts check box if you want to allow users other than root to mount file systems.
8. Deselect the Internet Address Verification check box if you do not want the `mountd` daemon to verify the IP address of each host requesting a mount or unmount. Internet Address Verification is enabled by default.
9. Select the Internet Address Verification & Domain Checking check box to have the `mountd` daemon verify that the host requesting a mount or unmount is in the server's domain.
10. Select the Internet Address Verification & Subdomain Checking check box to have the `mountd` daemon verify that the host requesting a mount or unmount is in the server's subdomain.
11. Specify the directories you want to export by following steps 2 through 7 in Section 4.5.2.
12. Select OK to validate your changes. The utility prompts you to start the NFS daemons.
13. Select Yes to save your configuration, start the daemons, and apply the changes immediately; or select No to save your configuration, close the Configure NFS Server dialog box, and apply the changes the next time you reboot your system.

    If you choose Yes, you are informed that the NFS daemons have been started. Select OK to dismiss the message and to close the Configure NFS Server dialog box.

You can also modify or deconfigure your server configuration after the initial setup. See the online help and Section 4.4 for more information.

4.3.2 Configuring an NFS Client

To configure an NFS client, do the following:

1. From the SysMan Menu, select Networking→Additional Network Services→Network File System (NFS)→Configure system as an NFS client. The Configure NFS Client dialog box is displayed.

   Alternatively, enter the following command on a command line:

   ```
   # /usr/bin/sysman nfs_client
   ```
2. Enter the number of client I/O threads to be run in the appropriate field.
3. Select the Enable Locking check box to specify locking configuration if the status of the `lockd` daemon is Stopped. If the status of the daemon is Running, locking is already set.
4. Select the Enable Automount Daemon check box to configure the `automount` daemon. See Section 4.1.2 for information on Automount

and Appendix A for information on Automount maps. If you would like to configure the AutoFS daemon, see Section 4.6.3.2 for more information.

5. Enter appropriate arguments to the `automount` daemon in the Automount Arguments field. See Section 4.6.3.4 for more information.
6. Specify the directories you want to import, those not already imported by `automount`, by following steps 2 through 10 in Section 4.6.1.
7. Select OK to validate the changes. (Due to the many `automount` arguments available, the validation of these arguments is deferred until the `automount` daemon starts and verifies them.)

   You are asked if you would like to start or restart the NFS daemons.
8. Select Yes to save the configuration, start the daemons, and apply your changes immediately; or select No to save the configuration, close the Configure NFS Client dialog box, and apply the changes the next time you reboot your system.

   If you choose Yes, you are informed that the NFS daemons have been started. Select OK to dismiss the message and to close the Configure NFS Client dialog box.

You can also modify or deconfigure your client configuration after the initial setup. See the online help and Section 4.4 for more information.

4.4 Deconfiguring NFS

You can use the SysMan Menu to deconfigure NFS servers and clients. When you deconfigure an NFS server or an NFS client, the corresponding NFS daemons stop and all of the corresponding NFS configuration information is deleted from the system. This action cannot be undone. To restore your NFS server or client, you must configure it again using the SysMan Menu.

When you deconfigure an NFS server, the client services are not removed. Likewise, when you deconfigure an NFS client, the server configuration is not removed. If you would like to deconfigure both the client and server configurations on a system, you must perform each action independently.

To deconfigure an NFS server, select Deconfigure system as an NFS Server from the SysMan Menu, or enter the following command on the command line:

```
# /usr/sbin/sysman nfs_deconfig_server
```

To deconfigure an NFS client, select Deconfigure system as an NFS Client from the SysMan Menu, or enter the following command on the command line:

```
# /usr/sbin/sysman nfs_deconfig_client
```

For both client and server, the Deconfigure NFS dialog box is displayed. Select Yes to deconfigure the service. You are informed that the service has been deconfigured. Select OK to dismiss the message and to close the dialog box.

4.5 Managing an NFS Server

This section describes how to perform the following NFS server tasks:

- Export a directory or file system
- Halt export of a directory or file system
- Enable a superuser on a client system to access files as superuser
- Send mail to superuser (root) across NFS
- Enable port monitoring
- Monitor the NFS load

4.5.1 Export Guidelines

The `/etc/exports` file defines an export list for each file system and directory that a client can mount. When creating entries in the `/etc/exports` file, remember the following:

- Make only one entry for each exported file system or directory; multiple entries are not supported.
- Each entry exports that directory and all subdirectories in it, except for those subdirectories that reside in a file system (disk partition) different from the exported directory.
- File systems and directories are exported with read-write access by default.
- If no remote system (client) names are specified for a file system or directory, any client on the network can mount that file system or directory.
- If one or more client names are specified for a file system or directory, only those clients can mount the exported file system or directory.
- If you are exporting a file system to a client that has multiple network interfaces on a subnet, you must specify the host names for all of the interfaces.
- If you start the `mountd` daemon with the `–i` option, only those hosts in the server's host database are allowed mount access. If you start the `mountd` daemon with the `–d` or `–s` option, only those clients in the same domain or subdomain, respectively, are allowed mount access.

- Exporting specific directories to specific clients provides more security than does exporting an entire file system to all clients.
- Protect sensitive exported data on the server by making the data files owned and accessible only by root, and do not allow superusers on client systems root access over NFS.
- The `-public` option can be specified by only one exported file system.

4.5.2 Exporting a File System or Directory

Exporting a file system or directory makes it available for client systems on the network to mount remotely. If you want your system to be an NFS server and to export file systems and directories, be aware that your system will be less secure. However, depending on how you export your files, you can minimize the security risks.

To export a file system by using the SysMan Menu, do the following:

1. From the SysMan Menu, select Networking→Additional Network Services→Network File System (NFS)→Configure system as an NFS server to display the Configure NFS Server dialog box.

   Alternatively, enter the following command on a command line:

   ```
   # /usr/bin/sysman nfs_server
   ```

2. Select the Shared Local Directories button to display the Share Local Directory dialog box.
3. Select Add to add a shared directory. The Add/Modify dialog box is displayed.
4. Enter the full path name of the directory to be exported in the Share this Directory field.
5. Select whether the directory has read/write or read-only access and whether all hosts or only selected hosts can have access. By default, the directory is exported with read/write permissions to all hosts.

   If you choose Selected in either the Read/Write or Read-Only dialog box, enter the name of each host that can have access to this directory in the appropriate field. Select Add for each host.
6. Select OK to validate the entry and to close the Add/Modify dialog box. Repeat steps 3 through 6 for additional directories.
7. Select OK to save the list of directories you chose to export in the `/etc/exports` file. You are informed that the changes have been made. Select OK to dismiss the message and to close the Share Local Directory dialog box.
8. Select OK to close the NFS Server dialog box.

You can also modify and delete exported directories with the Share Local Directory dialog box. See Section 4.5.3 and the online help for more information.

Optionally, you can use a text editor to add, modify, and delete exported directories directly in the `/etc/exports` file. See `exports`(4) for more information about editing this file.

4.5.3 Halting Export of a Directory or File System

Halting export of a directory or file system prevents client systems from accessing the particular directory or file system; you can still export other directories or file systems. If you do not want to export any file systems, you might want to deconfigure your NFS server as documented in Section 4.4.

To halt the export of a file system by using the SysMan Menu, do the following:

1. From the SysMan Menu, select Networking→Additional Network Services→Network File System (NFS)→Configure system as an NFS server to display the Configure NFS Server dialog box.

   Alternatively, enter the following command on a command line:

   ```
   # /usr/bin/sysman nfs_server
   ```

2. Select the Shared Local Directories button to display the Share Local Directory dialog box.
3. Select the entry that you no longer want to export from the list of shared directories.
4. Select Delete to remove the highlighted entry from the list. Repeat steps 3 and 4 to halt the export of additional entries.
5. Select OK to save the remaining list of exports in the `/etc/exports` file. You are informed that the changes have been made. Select OK to dismiss the message and to close the Share Local Directory dialog box.
6. Select OK to close the NFS Server dialog box.

You can also add and modify exported directories with the Share Local Directory dialog box. See Section 4.5.2 and the online help for more information.

Optionally, you can use a text editor to add, modify, and delete exported directories directly in the `/etc/exports` file. See `exports`(4) for more information about editing this file.

4.5.4 Enabling Client Superuser Access to Files

By default under NFS, a superuser (root) on a client system does not have superuser privileges on the server and cannot do the following:

- Access remotely mounted files and directories whose permissions do not allow world access
- Change the ownership of remotely mounted files (run the `chown` command)

For security reasons, it is best not to allow a remote superuser access to your system as superuser unless both the remote host and superuser are trusted. However, in a friendly network environment, you can explicitly allow superuser access over the network.

To allow a superuser on a client access to your server system, edit the `/etc/exports` file on your server and add the `–root=0` option to the entry you want to make available. The `–root=0` option maps the remote superuser's identification to UID 0. All future mount requests will be honored with root mapping. By default, this option allows superuser access from any client system on the network. To restrict the superuser access to specific systems, use the `–root=`*`host_list`* option, where *`host_list`* is a list of host names. See `exports`(4) for more information.

By default, NFS servers regard superusers and those users without UNIX authentication (personal computer systems) as anonymous users. This class of users can only access files that are accessible to the world. To prevent anonymous users from accessing file systems or directories, use the `–anon=–1` option. If you still want to allow client superusers access to the file systems or directories, specify the `–root` option in addition to the `–anon` option. The `–root` option overrides the `–anon` option for client superusers only.

A superuser on a client system can assume the identity of any other user on the client system by substituting the UID number. The client superuser could then have the access rights of another user on the server. Therefore, to protect sensitive exported data on the server, make root the owner of the data files and do not export the directory or file system with root mapping. This is useful if you need to export other files in the file system.

The following example shows entries in an `/etc/exports` file:

```
/usr/games –root=0 host8 [1]
/usr/templates –root=host8 [2]
```

[1] Exports the `/usr/games` file system. It can be mounted remotely (read-write) only by the client system host8. However, the client superuser has superuser access to the file system. The superuser's UID is 0 (zero).

2 Exports the /usr/templates file system. It can be mounted remotely (read-write) by any client in the network. However, only the superuser on host8 has superuser access to the file system.

4.5.5 Sending Mail to Superuser (root) Across NFS

If the /usr/spool/mail directory is remotely mounted from the server, and the directory is not exported with the root=0 option, client users will not be able to send mail to the superuser (root) on the server. To enable clients to send mail to root, set the root and admin aliases to the login name or names of the system administrators for that system. Then, users can address all mail intended for the administrators of that system as follows:

admin@*system*

To enable clients to send mail to root, follow these steps:

1. Edit the /var/adm/sendmail.cf file and add the alias name admin to the following line:

   ```
   CN MAILER-DAEMON postmaster
   ```

   The resulting line will look like the following line:

   ```
   CN MAILER-DAEMON postmaster admin
   ```

   This adds the name admin to the class N.

   Alternatively, you can run the Mail Configuration application and add admin as a local user. See Chapter 7 for more information.
2. Edit the /var/adm/sendmail/aliases file, add the login names of the system administrators, and redefine (alias) the name root to be admin.
3. Restart the sendmail daemon by using the following command:

   ```
   # /sbin/init.d/sendmail restart
   ```

If you are enabling clients to send mail to root, remember the following:

- It is best for all systems in the local area network (LAN) to follow this convention. Mail for root or admin on any system can be automatically directed to any user login on any system.
- A /usr/spool/mail/root mailbox is not created or used.

The following example shows the steps involved in enabling clients to send mail to root:

```
# vi /var/adm/sendmail/sendmail.cf 1
   .
   .
   .
# vi /var/adm/sendmail/aliases 2
   .
   .
   .
```

```
# /sbin/init.d/sendmail restart 3
```

1 Opens the /var/adm/sendmail/sendmail.cf file to add the admin alias.

2 Opens the /var/adm/sendmail/aliases file to add the login names and root alias.

3 Restarts the sendmail daemon.

The following example shows entries in the /var/adm/sendmail/aliases file for the system administrators John, Mary, and Joe:

```
admin:john,mary,joe
root:admin
```

4.5.6 Enabling Port Monitoring

Only privileged users can attach to Internet domain source ports known as privileged ports. By default, NFS does not check to see if a client is bound to a privileged port. You might want to activate NFS server port monitoring to be sure that file access requests are generated by the client kernel rather than forged by an application program.

Although this operating system enforces the privileged port convention, some operating systems do not. If hosts running a different operating system are on your network, activating port checking might not improve security, but could prevent those systems from functioning properly as NFS client systems.

To start NFS server port monitoring, enter the following command:

```
# /usr/sbin/nfsportmon on
```

To stop source port monitoring, enter the following command:

```
# /usr/sbin/nfsportmon off
```

4.5.7 Monitoring the NFS Load

Monitoring the NFS load allows you to see the number of NFS requests, both client and server, being executed on the local machine. It is a good idea to monitor NFS requests periodically to determine whether you need additional NFS server threads.

To monitor NFS requests, use the nfsstat command with the following syntax:

nfsstat –**n**

See nfsstat(8) for more information on monitoring NFS load.

The following example shows the client and server activity on a local machine:

```
# /usr/bin/nfsstat –n
 nfs:
calls      badcalls
69228      0

Server nfs V2:
null       getattr    setattr    root       lookup     readlink   read
1  0%      24 0%      0  0%      0  0%      60  0%     0  0%      5  0%
wrcache    write      create     remove     rename     link       symlink
0  0%      58030 83%  20  0%     0  0%      0  0%      0  0%      0  0%
mkdir      rmdir      readdir    statfs
0  0%      0  0%      0  0%      2  0%

Server nfs V3:
null       getattr    setattr    lookup     access     readlink   read
0  0%      667  0%    1009  1%   2598  3%   101  0%    200  0%    1408  2%
write      create     mkdir      symlink    mknod      remove     rmdir
1280  1%   376  0%    71  0%     200  0%    0  0%      676  0%    70  0%
rename     link       readdir    readdir+   fsstat     fsinfo     pathconf
100  0%    100  0%    468  0%    0  0%      1750  2%   2  0%      0  0%
commit
10  0%

Client nfs:
calls      badcalls   nclget     nclsleep
224664     0          224664     0

Client nfs V2:
null       getattr    setattr    root       lookup     readlink   read
0  0%      51328 22%  1069  0%   0  0%      41643 18%  455  0%    28793  12%
wrcache    write      create     remove     rename     link       symlink
0  0%      64665 28%  589  0%    1052  0%   352  0%    250  0%    250  0%
mkdir      rmdir      readdir    statfs
171  0%    170  0%    2689  1%   1814  0%

Client nfs V3:
null       getattr    setattr    lookup     access     readlink   read
0  0%      2038  0%   2180  0%   8534  3%   430  0%    450  0%    3136  1%
write      create     mkdir      symlink    mknod      remove     rmdir
3158  1%   1048  0%   243  0%    450  0%    1  0%      1848  0%   242  0%
rename     link       readdir    readdir+   fsstat     fsinfo     pathconf
452  0%    350  0%    1240  0%   0  0%      3506  1%   3  0%      0  0%
commit
75  0%
```

4.6 Managing an NFS Client

Your system can be an NFS client if the following conditions exist:

- Your system can reach an NFS server over the network.
- Your system's host or network group name is included in the server's `/etc/exports` file, or the server is exporting a file system to all systems on the network.

This section describes how to perform the following NFS client tasks:

- Mount a remote file system or directory

- Mount a remote file system or directory with Automount or AutoFS
- Unmount a remote file system or directory

4.6.1 Mounting a Remote File System or Directory

You can mount a remote file system or any subdirectory within a remote file system onto a local mount point. While mounted, it is treated as a file system by the local system.

To mount a remote file system or directory by using the SysMan Menu, do the following:

1. From the SysMan Menu, select Networking→Additional Network Services→Network File System (NFS)→Configure system as an NFS client to display the Configure NFS Client dialog box.

   Alternatively, enter the following command on a command line:

   ```
   # /usr/bin/sysman nfs_client
   ```

2. Select the Mount Network Directories button to display the Mount Network Directory dialog box.

   A list of NFS-mounted directories that are saved in the `/etc/fstab` file is displayed. Remote file systems that you mounted by using the `mount` command are not included in this list.

3. Select Add to add a remote directory. The Add/Modify dialog box is displayed.
4. Enter the host name of the NFS server from which the remote directory is exported in the Remote Host Name field.
5. Enter the full path name of the directory to be imported in the Remote Directory Path field.
6. Enter the full path name of the local directory on which the imported directory is to be mounted in the Local Mount Point field.
7. Select whether the directory has read-only or read/write access with the appropriate radio button.
8. Select the Mount on Reboot checkbox if you want the directory to be mounted each time you reboot.
9. Select OK to validate the entry and to close the Add/Modify dialog box. Repeat steps 3 through 9 for additional directories.
10. Select OK to save the list of directories you chose to import. The names of those directories that are to be mounted on reboot are saved in the `/etc/fstab` file.

    You are informed that the changes have been made. Select OK to dismiss the message and to close the Mount Network Directory dialog box.

11. Select OK to close the NFS Client dialog box.

You can also modify and delete your imported directories with the Mount Network Directory dialog box. See Section 4.6.2 and the online help for more information.

Each directory imported via the Mount Network Directory dialog box is mounted using the `bg` and `hard` options of the `mount` command. If the first attempt to mount the directory fails, the client tries mounting it in the background (`bg` option), and it continues attempting to mount the directory until the server responds (`hard` option). No other `mount` options can be selected via the dialog box.

Optionally, you can use the `mount` command to mount remote file systems from the command line. Or, you can use a text editor to directly add, modify, or delete entries in the `/etc/fstab` file. Use these alternatives if you need to specify `mount` options that are not supported by the Mount Network Directory dialog box. See `mount`(8), `umount`(8), and `fstab`(4) for more information.

4.6.2 Unmounting a Remote File System or Directory

Unmounting a remote file system or directory removes access to a particular file system or directory that is being imported from an NFS server; you can still import other directories or file systems. If you do not want to import any file systems, you might want to deconfigure your NFS client as documented in Section 4.4.

To unmount a remote file system or directory by using the SysMan Menu, do the following:

1. From the SysMan Menu, select Networking→Additional Network Services→Network File System (NFS)→Configure system as an NFS client to display the Configure NFS Client dialog box.

   Alternatively, enter the following command on a command line:

   ```
   # /usr/bin/sysman nfs_client
   ```

2. Select the Mount Network Directories button to display the Mount Network Directory dialog box.

   A list of NFS-mounted directories that are saved in the `/etc/fstab` file is displayed. Remote file systems that you mounted by using the `mount` command are not included in this list. Use the `umount` command to unmount these file systems. See `umount`(8).

3. Select the entry that you want to unmount from the list.

4. Select Delete to remove the highlighted entry from the list. Repeat steps 3 and 4 to remove additional entries

5. Select OK to save the current list of imported directories in the `/etc/fstab` file.

   You are informed that the changes have been made. Select OK to dismiss the message and to close the Mount Network Directory dialog box.

6. Select OK to close the NFS Client dialog box.

You can also add and modify your imported directories with the Mount Network Directory dialog box. See Section 4.6.1 and the online help for more information.

Optionally, you can use the `umount` command to unmount remote file systems from the command line. Or, you can use a text editor to directly add, modify, or delete entries in the `/etc/fstab` file. See `mount`(8), `umount`(8), and `fstab`(4) for more information.

4.6.3 Automatically Mounting a Remote File System

The following sections describe how to configure Automount and AutoFS, two services that allow you to automatically mount a remote file system or directory at the time of access.

Note

The Automount daemon will be retired in a future release of the operating system. For information about migrating from Automount to AutoFS, see Section 4.6.3.5.

Before starting the configuration procedure for either service, determine whether or not you are using local maps or NIS-distributed maps. See Section 4.1.2 for a description of local and NIS-distributed maps.

4.6.3.1 Using Automount to Mount a Remote File System

To use local Automount maps, do the following:

1. Log in as root.
2. Create a local `auto.master` map. You can create this and other maps in any directory on the system, but they are conventionally located in the `/etc` directory, where the SysMan Menu expects to find them.

   See Appendix A for information on creating maps.

Note

If you are modifying an existing `auto.master` map, you must stop and restart the `automount` daemon to apply the revised map.

3. Create the local maps for your system.
4. Start the `automount` daemon by using the NFS Client dialog box of the SysMan Menu. See Section 4.3.2 for information on starting the `automount` daemon.

When the `automount` daemon starts, it uses the local `auto.master` file to determine the location of other maps, their local mount points, and the mount options.

To use NIS-distributed Automount maps, do the following:

1. Set up your system as an NIS client. See Section 3.3.3 for information on setting up an NIS client.
2. Start the `automount` daemon by using the NFS Client dialog box of the SysMan Menu. See Section 4.3.2 for information on starting the `automount` daemon.

The NIS master server serves all Automount maps in the domain. When the `automount` daemon starts, it uses the master `auto.master` file to determine the location of other maps, their local mount points, and the mount options.

If you alter your local or NIS-distributed Automount maps at any time, you must restart the `automount` daemon on clients as follows to apply the changes:

1. From the SysMan Menu, select Networking→Additional Network Services→Network File System (NFS)→Configure system as an NFS client to display the Configure NFS Client dialog box.

   Alternatively, enter the following command on a command line:

   ```
   # /usr/bin/sysman nfs_client
   ```

2. Deselect the Enable Automount check box.
3. Select OK to disable Automount and Yes to restart the NFS daemons. A message indicates that the daemons are restarted; select OK to dismiss the message and close the NFS Client dialog box.
4. Open the NFS Client dialog box again.
5. Select the Configure for Automount check box.

6. Select OK to enable Automount and Yes to restart the NFS daemons. A message indicates that the daemons are restarted.
7. Select OK to dismiss the message and to close the NFS Client Setup dialog box.

See automount(8) for information on the automount command and its arguments.

4.6.3.2 Using AutoFS to Mount a Remote File System

To use local AutoFS maps, do the following:

1. Log in as root.
2. Create a local auto.master map. You can create this and other maps in any directory on the system, but they are conventionally located in the /etc directory.

   See Appendix A for information on creating maps.

   Note

   If you are modifying an existing auto.master map, you must process the map with the autofsmount command to apply the changes. See Section 4.6.3.3 for more information.

3. Create the local maps for your system.
4. Start the autofsd daemon by entering the following command:

   # **/usr/sbin/autofsd**

5. Execute the autofsmount command to process your local master file:

   # **/usr/sbin/autofsmount -m -f *local_master_file***

6. Use the rcmgr utility to configure AutoFS to start each time you boot your system. The AutoFS parameters in the following steps are case sensitive and must be typed in uppercase as shown.

   a. Enable the AutoFS daemon by entering the following command:

      # **rcmgr -c set AUTOFS 1**

   b. Specify arguments for the autofsd daemon and the autofsmount command, as follows:

      # **rcmgr -c set AUTOFSD_ARGS "*arguments*"**
      # **rcmgr -c set AUTOFSMOUNT_ARGS "-m -f *local_master_file*"**

      You must define the AUTOFSD_ARGS parameter, even if you do not need to define arguments for the autofsd daemon. If there are no arguments, specify open and close quotation marks (“”).

See autofsd(8) and autofsmount(8) for information about valid arguments.

When the autofsmount command is executed, it installs intercept points into the kernel based on the maps you created. When users access the associated file systems, the kernel communicates with the autofsd daemon to mount and unmount the file systems based on the map entries.

To use NIS-distributed AutoFS maps, do the following:

1. Set up your system as an NIS client. See Section 3.3.3 for information on setting up an NIS client.
2. Start the autofsd daemon by entering the following command:

   ```
   # /usr/sbin/autofsd
   ```

3. Execute the autofsmount command to process the NIS-distributed master file:

   ```
   # /usr/sbin/autofsmount
   ```

   When you execute the autofsmount command with no arguments, it automatically processes the NIS-distributed auto.master file.

4. Use the rcmgr utility to configure AutoFS to start each time you boot your system. The AutoFS parameters in the following steps are case sensitive and must be typed in uppercase as shown.

   a. Enable the AutoFS daemon by entering the following command:

      ```
      # rcmgr -c set AUTOFS 1
      ```

   b. Specify arguments for the autofsd daemon and the autofsmount command, as follows:

      ```
      # rcmgr -c set AUTOFSD_ARGS "arguments"
      # rcmgr -c set AUTOFSMOUNT_ARGS ""
      ```

      You must define the AUTOFSD_ARGS parameter, even if you do not need to define arguments for the autofsd daemon. If there are no arguments, specify open and close quotation marks (“”).

      Setting the autofsmount command to run with no arguments indicates that AutoFS is to use the NIS-distributed auto.master file.

      See autofsd(8) and autofsmount(8) for information about valid arguments.

The NIS master server serves all AutoFS maps in the domain. When the autofsmount command is executed, it uses the master auto.master file to determine the location of other maps, their local mount points, and the mount options.

See autofsd(8) and autofsmount(8) for more information. See sys_attrs_autofs(5) for tuning information and Section 9.9 for troubleshooting information.

4.6.3.3 Modifying Your AutoFS Configuration

If you alter your local or NIS-distributed AutoFS maps at any time, you must process the affected maps with the autofsmount command to apply the changes. If you need to add, modify, or remove an AutoFS mount, do the following:

1. Edit the affected maps to add, modify, or remove the appropriate entries. If you use NIS to distribute AutoFS maps, see Section 3.4.5 for information about distributing your updated maps.
2. If you have manually mounted a file system on the target directory for any AutoFS mount that you intend to add, remove these mounts as described in Section 4.6.2 or mount(8). AutoFS cannot auto-mount a file system on a mount point that is occupied by an active NFS file system.
3. If you have modified or removed any existing AutoFS map entries, you must remove the corresponding mount or symbolic link for each of these entries on your client system.

   As described in Section A.4, AutoFS can serve a file system by mounting it directly on its intended mount point, if the file system is on a remote system, or by creating a symbolic link, if the file system is located on the local system. If the auto-mounted file system is served from a remote system, you can remove it by executing the following command:

   ```
   # /usr/sbin/autofsmount -t directory
   ```

   If the auto-mounted file system is served by the local system through a symbolic link, you can remove it by executing the rm command, as follows:

   ```
   # rm link
   ```

4. Execute the autofsmount command with the appropriate arguments to process the new AutoFS map or maps. If you defined the arguments in the AUTOFSMOUNT_ARGS parameter of the rc.config.common file, as described in Section 4.6.3.2, you can execute the following command:

   ```
   # /usr/sbin/autofsmount 'rcmgr -c get AUTOFSMOUNT_ARGS'
   ```

   See Section 4.6.3.4 and autofsmount(8) for more information about specifying autofsmount arguments.

4.6.3.4 Specifying automount and autofsmount Arguments

You can specify arguments for the automount or autofsd daemons from the command line, in a local auto.master map, in an NIS-distributed

auto.master map, or some combination of the three. However, it is important to know that the daemons read and carry out their instructions in the following order:

1. Command line information, such as additional mount points or replacements to entries in a master map, are read first. Command line information takes precedence over instructions in any maps — local or NIS-distributed.
2. Instructions in a local auto.master map (specified with the –f option) are read next. The information in the local master map overrides information in an NIS-distributed master map.
3. Information in the NIS-distributed master map is read last.

When you invoke the automount or autofsmount commands without any arguments, they look for a distributed NIS map called auto.master. If they find one, the commands check the master map for information about the location of other maps, their local mount points, and the mount options. If they do not find one, and if no local auto.master map is specified, the commands exit.

You can pass command arguments to the automount daemon from the NFS Client dialog box of the SysMan Menu as documented in Section 4.3.2. You can also pass arguments to either the automount or autofsmount command in one of the following ways:

- Specify all of the arguments to either command on the command line. For example:

```
# automount /net –hosts \
  /home /etc/auto.home –rw,intr \
  /- /etc/auto.direct –ro,intr
```

- Specify all of the arguments to either command in the rc.config.common file by using the rcmgr utility. Arguments you specify in this file are passed to the command when you boot your system. For example:

```
# rcmgr -c set AUTOMOUNT_ARGS "/net –hosts \
  /home /etc/auto.home –rw,intr \
  /- /etc/auto.direct –ro,intr"
```

  To define command arguments for the autofsmount command, use the AUTOFSMOUNT_ARGS parameter, as follows:

```
# rcmgr -c set AUTOFSMOUNT_ARGS "/net –hosts \
  /home /etc/auto.home –rw,intr \
  /- /etc/auto.direct –ro,intr"
```

- Include the arguments from the previous examples in an NIS-distributed auto.master map:

```
/net    –hosts
/home   /etc/auto.home          –rw,intr
/-      /etc/auto.direct        –ro,intr
```

If this NIS `auto.master` map is distributed, typing the `automount` or `autofsmount` command at the superuser prompt (#) produces the same results as the previous command line.

- Include the arguments in a local `auto.master` file and use the –`f` option to instruct the `automount` or `autofsmount` command to process the local `auto.master` file. The –`f` option instructs these commands to consult the local master map first and then the NIS-distributed master map. For example:

  ```
  # automount –f /etc/auto.master
  ```

  You can also add the –`m` option, which forces the commands to ignore the NIS-distributed master map, if there is one. For example:

  ```
  # automount –m –f /etc/auto.master
  ```

- Specify mount points on the command line, in addition to those included in the local `auto.master` file. For example:

  ```
  # automount –f /etc/auto.master \
    /src /etc/auto.src –ro,soft
  ```

- Nullify one of the entries in the local `auto.master` map. For example:

  ```
  # automount –f /etc/auto.master /home –null
  ```

  This option is currently not supported by the `autofsmount` command.

- Replace an entry in the local `auto.master` map with one of your own. For example:

  ```
  # automount –f /etc/auto.master \
   /home /mine/auto.home –rw,intr
  ```

AutoFS provides one additional convenience for complicated configurations that require many cumbersome `autofsmount` arguments. It allows you specify your `autofsmount` arguments in an environment variable called `AUTOFSMOUNT_EXPARGS`, which is subsequently imported by the `autofsmount` command when you invoke it with the `-e` option. You can arrange for this by adding the appropriate statements to the configuration file for your shell, as follows.

For C Shell (in `.cshrc` file):

```
setenv AUTOFSMOUNT_EXPARGS  ‘rcmgr -c get AUTOFSMOUNT_ARGS‘
```

For Korn Shell (in `.profile` file):

```
AUTOFSMOUNT_EXPARGS=‘rcmgr -c get AUTOFSMOUNT_ARGS‘
export AUTOFSMOUNT_EXPARGS
```

With the environment variable set in this manner, you can invoke the `autofsmount` command with all of your predefined options as follows:

```
# autofsmount -e
```

If necessary, you can include additional options after the `-e` option.

See `automount`(8) and `autofsmount`(8) for more information on these commands and their arguments.

4.6.3.5 Migrating from Automount to AutoFS

Automount will be retired in a future release; therefore, if you are using Automount, you will eventually need to migrate to AutoFS. If you want to migrate now, you can use the instructions in this section to complete the task. (If your system is a node in a cluster, see the TruCluster Server *Cluster Administration* manual for information about migrating from Automount to AutoFS in a cluster environment.)

There are two procedures for migrating from Automount to AutoFS. The primary difference between the procedures is that the first one requires you to reboot the operating system, and the second one allows the system to remain in multi-user mode. The first procedure is recommended because it is less complicated and it ensures that all automounted file systems will be cleanly unmounted, but the second procedure is provided as an alternative for high-availability systems that cannot be rebooted.

It is recommended that you familiarize yourself with AutoFS by reading Section 4.1.2, Section 4.6.3, `autofsd`(8), and `autofsmount`(8) before proceding with these migration steps.

Note that Automount and AutoFS maps are compatible, with the few exceptions that are mentioned in the Restrictions section of `autofsmount`(8).

4.6.3.5.1 Recommended Migration Path

To migrate from Automount to AutoFS on a system that can be rebooted, do the following:

1. Determine the arguments that you must pass to the `autofsmount` command at boot time. See the `autofsmount` reference page for a list of valid options and arguments.

   Note that these arguments are typically a subset of those you have already specified for the `automount` daemon in the `AUTOMOUNT_ARGS` parameter of the `/etc/rc.config.common` file. To view that parameter, execute the following command:

   ```
   # /usr/sbin/rcmgr -c get AUTOMOUNT_ARGS
   ```

To set the corresponding arguments for the autofsmount command, define the AUTOFSMOUNT_ARGS parameter as follows:

```
# /usr/sbin/rcmgr -c set AUTOFSMOUNT_ARGS "arguments"
```

2. Determine the arguments that you must pass to the autofsd daemon at boot time. See the autofsd reference page for a list of valid options and arguments.

   These arguments are typically any environment variable definitions (-D option) that you have already specified for the automount daemon in the same AUTOMOUNT_ARGS parameter that was mentioned in step 1.

   To set the corresponding arguments for the autofsd daemon, define the AUTOFSD_ARGS parameter as follows:

   ```
   # /usr/sbin/rcmgr -c set AUTOFSD_ARGS "arguments"
   ```

   If you have not defined any environmental variables with the automount daemon, it is possible that there will be no autofsd arguments. In this case, you must specify the AUTOFSD_ARGS parameter with open and close quotation marks (“”).

3. Disable Automount and enable AutoFS in the /etc/rc.config.common file by entering the following commands:

   ```
   # /usr/sbin/rcmgr -c set AUTOMOUNT 0
   # /usr/sbin/rcmgr -c set AUTOFS 1
   ```

   These values specify that AutoFS will be the only automatic mounting service in effect when the system boots.

4. Reboot the system to put the new AutoFS configuration into effect.

   Note that Automount will remain fully functional until you reboot. Do not attempt to start AutoFS while Automount is running.

4.6.3.5.2 High-Availability Migration Path

To migrate from Automount to AutoFS on a system that cannot be rebooted, do the following:

1. Follow steps 1-3 in Section 4.6.3.5.1.
2. Release all auto-mounted file systems by eliminating their use.

   In terminal windows, if you are operating in an automounted file system, change directory into another file system. If you have any files open in an automounted file system, close the files and, if necessary, close any applications that might have a lock on these files.

3. Stop the automount daemon.

   a. Determine the process IDs of all Automount tasks by entering the following command:

```
# ps -ef | grep automount
```

b. Stop each Automount process by entering the following command:

```
# kill -SIGTERM process-ID
```

4. Verify that automounted file systems are no longer mounted by entering the following command:

```
# /sbin/mount -e | grep temporary_mount_dir
```

Replace *temporary_mount_dir* with the name of the directory in which Automount temporarily mounts file systems. By default, this directory is `tmp_mnt`.

This command searches for instances of *temporary_mount_dir* in the current list of all mount points. If any such mount points still exist, they will still be usable under their *temporary_mount_dir* pathnames, but they will no longer be usable via the expected pathnames under which they were served by Automount. If these mount points later become idle, you can remove them by entering the following command:

```
# /sbin/umount -f mount-point
```

None of these mounts will affect AutoFS, because AutoFS does not use a temporary mount directory.

5. Start the AutoFS service with the arguments you previously specified in the `/etc/rc.config.common` file.

a. Start the `autofsd` daemon, as follows:

```
# /usr/sbin/autofsd arguments
```

To display the arguments you already defined for `autofsd`, enter the following command:

```
# /usr/sbin/rcmgr -c get AUTOFSD_ARGS
```

b. Invoke the `autofsmount` command, as follows:

```
# /usr/sbin/autofsmount arguments
```

To display the arguments you already defined for `autofsmount`, enter the following command:

```
# /usr/sbin/rcmgr -c get AUTOFSMOUNT_ARGS
```

If you properly set the AutoFS arguments in the `/etc/rc.config.common` file, you will not need to start AutoFS in this manner again. It will automatically start each time that you boot the operating system.

5

UNIX-to-UNIX Copy Program

The UNIX-to-UNIX Copy Program (UUCP) is a group of programs that enables batched, error-free file transfer and remote command execution between two UNIX systems. UUCP is typically used to transfer electronic mail, network nets, and public domain software over low-speed, low-cost communications links. Tru64 UNIX implements the HoneyDanBer version of UUCP.

This chapter describes:

- The UUCP environment (Section 5.1)
- How to plan for your UUCP configuration (Section 5.2)
- How to configure your system for UUCP (Section 5.3)
- How to manage UUCP (Section 5.4)

For general information about UUCP see `uucp_intro`(7). For information on how to use UUCP, see the *Command and Shell User's Guide*.

For troubleshooting information, see Section 9.10.

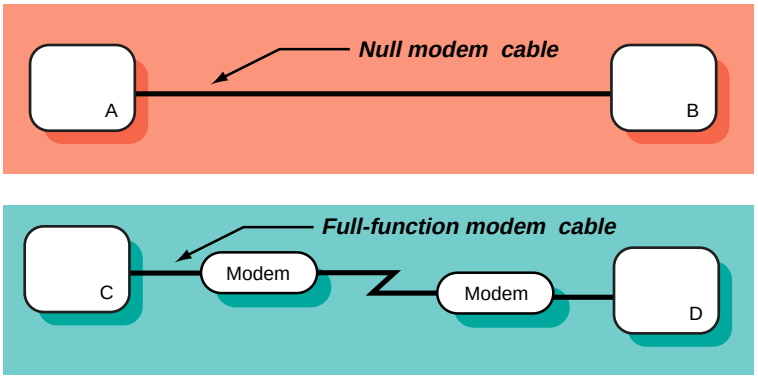
5.1 UUCP Environment

In the UUCP environment, systems can be connected to each other in the following ways:

- Directly connected to each other, if they are in close proximity
- Connected through modems and a telephone network, if they are not in close proximity
- Connected through a local area network (LAN), if they are not in close proximity

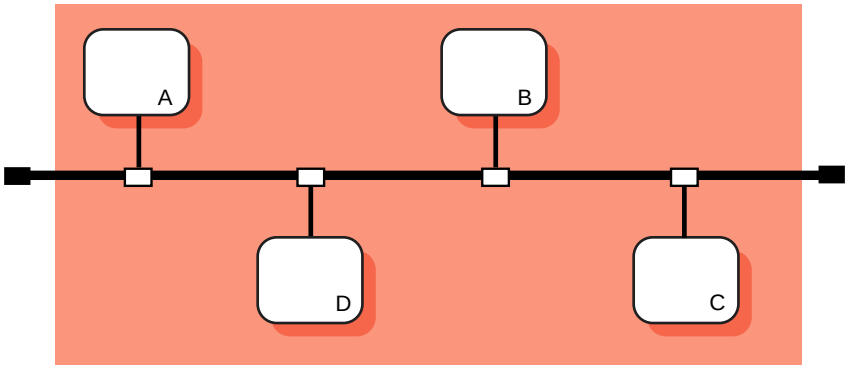
Figure 5–1 shows two simple UUCP configurations. Figure 5–2 shows a sample UUCP configuration on a LAN in which Host A has a TCP/IP connection with Host C.

Figure 5–1: Sample Simple UUCP Configuration



ZK-1174U-AI

Figure 5–2: Sample UUCP Over TCP/IP Configuration



ZK-1175U-AI

5.2 Planning UUCP

This section describes those tasks you need to do before configuring UUCP.

5.2.1 Verifying the Correct Hardware

In verifying the correct hardware, you need to verify both the cables and modems, if used.

Make sure you are using the correct cable to connect to the serial port of your system. If you do not, you might experience signal degradation and the software will fail to function properly.

See the Point-to-Point Connections chapter in *Network Administration: Connections* for a list of modem cables to use. If the two systems are in close proximity to each other, use one of the null modem cables. If the two systems are connected through modems and telephone lines, use a standard modem cable. When using modems with UUCP, make sure that both the local and the remote modems are correctly configured.

UUCP can also be configured to run over TCP/IP local area networks (LANs). For information on running UUCP over a LAN, see `uucp_manual_setup`(7).

5.2.2 Preparing for the Configuration

UUCP configuration consists of defining the following parts:

- Connection information for your system
- Dial-up information for outgoing calls
- Information for receiving incoming calls

The type of information you need depends on the types of connections you plan to set up and use. The following sections contain worksheets that you can use to record the information required to configure UUCP.

5.2.2.1 Information for Connections

Figure 5–3 shows the UUCP Setup Worksheet. If you are viewing this manual online, you can use the print feature to print a copy of the worksheet. The sections that follow explain the information you need to record on the worksheet.

Figure 5–3: UUCP Setup Worksheet

UUCP Setup Worksheet

Type of connection: ☐ Modem ☐ Direct link ☐ TCP/IP

Modems:

Modem type: _____

Baud rate: _____ ☐ Any

Device name: _____

/etc/inittab entry ID: _____

Direct links:

Remote system name: _____ ☐ Direct

Baud rate: _____ ☐ Any

Device name: _____

/etc/inittab entry ID: _____

TCP/IP:

Outgoing connections: ☐ Yes ☐ No

Incoming connections: ☐ Yes ☐ No

Type of connection

The types of connections you want to configure. You can configure one or all of the following connections:

- Modems — Modems enable you to use UUCP over analog transmission facilities, which include telephone lines.
- Direct (hardwired) links — Direct hardwired links connect systems with cables.
- TCP/IP — Connections using the TCP/IP protocol.

For modem connections, supply the following information:

Modem type

The type of modem you want to use. The supported devices are listed in the `/usr/lib/uucp/Devices` file. For more information, see `uucp_manual_setup`(7).

Baud rate or Any

The speed at which the modem is to operate; for example: 1200, 2400, 9600, or any.

Device name

The name of the tty device that you want the modem to use, as listed in the `/dev` directory. If you are unsure of the terminal device, see `ports`(7).

/etc/inittab entry ID

The process ID for the `uugetty` process entry in the `/etc/inittab` file. The `uugetty` process sets up speed, terminal flags, and the line discipline for managing terminals. For more information, see `uugetty`(8).

Note

Run the `uugetty` command only on RS-232 lines, not printer or console lines.

For direct link connections, supply the following information:

Remote system name or Direct

The type of direct link. If you want to connect to a specific remote system, enter the name of the remote system. This restricts connections to that system only.

If you want to connect to any system to which you have a direct hardwired connection, check Direct.

Baud rate or Any

The speed at which the direct link is to operate; for example: 1200, 2400, 9600, or any.

Device name

The name of the tty device that you want the direct link to use, as listed in the `/dev` directory. If you are unsure of the terminal device, see `ports`(7).

/etc/inittab entry ID

The process ID for the `uugetty` process entry in the `/etc/inittab` file. The `uugetty` process sets up speed, terminal flags, and the line discipline for managing terminals. For more information, see `uugetty`(8).

Note

Use the `uugetty` command to configure only RS-232 lines, not printer or console lines.

For TCP/IP connections, supply the following information:

Outgoing connections

If you want to configure UUCP to place outgoing calls over TCP/IP, check Yes. When you enable UUCP to place outgoing calls over TCP/IP, an entry for TCP/IP is added to the `/usr/lib/uucp/Devices` file.

Otherwise, check No.

Incoming connections

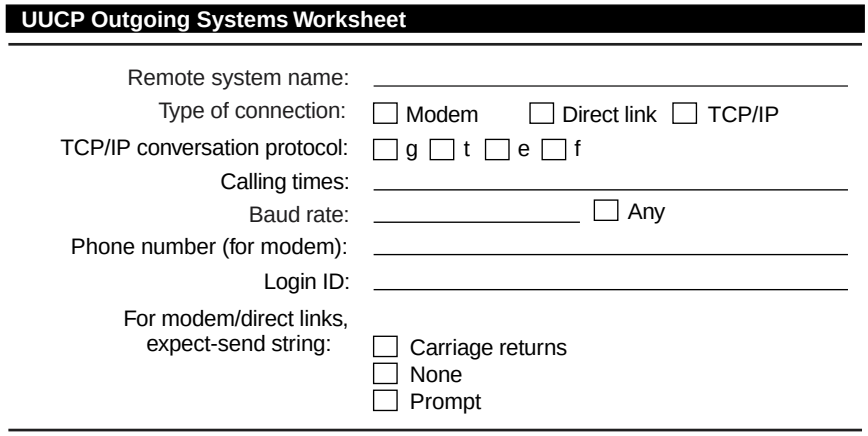
If you want to configure UUCP to accept incoming calls over TCP/IP, check Yes. When you enable UUCP to accept incoming calls over TCP/IP, the `/etc/inetd.conf` file is modified. In addition, you must stop and restart the `inetd` daemon to be able to accept UUCP calls over TCP/IP.

Otherwise, check No.

5.2.2.2 Information for Outgoing Systems

Figure 5–4 shows the UUCP Outgoing Systems Worksheet. If you are viewing this manual online, you can use the print feature to print a copy of the worksheet. The sections that follow explain the information you need to record on the worksheet.

Figure 5–4: UUCP Outgoing Systems Worksheet



UUCP Outgoing Systems Worksheet

Remote system name: _____

Type of connection: ☐ Modem ☐ Direct link ☐ TCP/IP

TCP/IP conversation protocol: ☐ g ☐ t ☐ e ☐ f

Calling times: _____

Baud rate: _____ ☐ Any

Phone number (for modem): _____

Login ID: _____

For modem/direct links, expect-send string:
☐ Carriage returns
☐ None
☐ Prompt

Remote system name

The name of the remote system to which you plan to connect.

Type of connection

The type of the connection. Check modem, direct hardwired, or TCP/IP. You must configure the type of the connection with the information from Section 5.2.2.1.

TCP/IP conversation protocol

For TCP/IP connections, the TCP/IP conversation protocol, which can be one of the following:

- `g` — Specifies the default protocol, which provides error checking.
- `t` — Presumes an error-free channel and therefore is not reliable for use with modem connections.
- `e` — Used to communicate with sites that are running both Tru64 UNIX and other UNIX versions of UUCP.
- `f` — Relies on flow control of the data stream. It is meant for working over links that can be guaranteed to be virtually error free, specifically X.25/PAD links.

Calling times

The times when your system is allowed to connect to the remote host. You can select the following times:

- Any time of any day

- Evenings — Monday to Friday 5 p.m. to 8 a.m.; Saturday and Sunday, all day
- Any three nights — You can choose the three nights from the following:
  - Monday to Friday, 11 p.m. to 8 a.m.
  - Saturday, all day
  - Sunday, until 5 p.m.
- Never

Baud rate or Any

The baud rate that corresponds to a device you configured in the `/usr/lib/uucp/Devices` file, or you can specify `any`, if the device can be used at any speed.

Phone number (for modem)

For modem connections, the telephone number of the remote system. You can enter the complete telephone number or a dialing prefix and the telephone number.

A dialing prefix is defined in the `/usr/lib/uucp/Dialcodes` file. The `/usr/lib/uucp/Dialcodes` file contains dial code abbreviations and partial phone numbers that complete the telephone entries in the `/usr/lib/uucp/Systems` file. Entries in the `/usr/lib/uucp/Dialcodes` file contain an alphabetic prefix attached to a partial phone number that can include, for example, access codes, area codes, and exchange numbers.

If you know the dialing prefix, enter it on the worksheet. If none is defined, enter it and the sequence of numbers to be associated with the prefix.

Login ID

The login name for your system on the remote system. This must match the information in the `/etc/passwd` file on the remote system. Ask the administrator of the remote system for the login name and password that is assigned to your system on the remote system. The administrator of the remote system must include the login name and password for your system in the remote system's `/etc/passwd` file.

Note

Although the password for the login ID on the remote system is required in order to configure UUCP, to protect system security do not write the password on this worksheet.

For modem/direct links, expect-send string

The *expect-send* string to be used immediately before performing the login on the remote system. You can choose one of the following:

- To send a series of carriage returns before expecting any characters from the remote system
- To specify no *expect-send* strings
- To be prompted to enter *expect-send* strings

Modems usually use a series of carriage returns as an *expect-send* string.

For more information on *expect-send* strings, see Systems(4).

5.2.2.3 Information for Incoming Systems

Figure 5–5 shows the UUCP Incoming Systems Worksheet. If you are viewing this manual online, you can use the print feature to print a copy of the worksheet. The sections that follow explain the information you need to record on the worksheet.

Figure 5–5: UUCP Incoming Systems Worksheet

UUCP Incoming Systems Worksheet

Remote system name: _____
Local system name: _____
Login ID: _____
Alternative login ID: _____
REQUEST option: ☐ Yes ☐ No
SENDFILES option: ☐ Yes ☐ Call
Additional READ/WRITE locations: _____
Additional NOREAD/NOWRITE locations: _____
Commands: _____
_____
_____
VALIDATE option: ☐ Yes ☐ No
CALLBACK option: ☐ Yes ☐ No
Phone number (for modem): _____

Remote system name

The name of the remote system you want to allow to establish incoming UUCP connections.

Local system name

The name of your system. The default provided is the name that you assigned to your system at installation.

Login ID

The login ID for the remote system. The login ID is automatically added to the `/etc/passwd` file on your system.

By convention, the login ID that you assign to a remote system establishing incoming connections is the system name prefixed with an uppercase u (U). For example, if you specify machine1 for incoming connections, the login ID, by convention, is Umachine1; however, you can select any login ID.

You also have the option of adding a comment to the `/etc/passwd` file for this login ID.

Alternative login ID

You have the option to assign more than one login ID for each incoming system. Assigning multiple logins to a remote system allows you to maintain better access control for users on the remote system. With multiple logins, you can grant privileged users on the remote system more access on your system than you do to nonprivileged users. With multiple logins, you can assign multiple sets of permissions.

You must provide this information to the administrator of each remote system that will connect to your system as an incoming system.

REQUEST option

If you want a remote system to ask for any queued work on the local system that is meant for that remote system, check Yes; otherwise, check No.

If you check Yes, remote system users can transfer files to and execute commands on a local system more easily. If security is a consideration, you can restrict this access so that the local system retains control of file transfers and command executions initiated by remote systems.

SENDFILES option

If you want the local system to try to send queued work to the calling remote system after the remote computer finishes transferring files to or executing commands on the local system, check Yes.

Security considerations at your site might require that you limit a remote system's access to the local system. In this case, check Call to send queued work to the remote system only when the local system contacts the remote system.

Additional READ/WRITE locations

If you do not specify pathnames in the READ and WRITE options, `uucp` permits files to be transferred only to the `/usr/spool/uucppublic` directory. However, if you specify pathnames in these options, you must enter the pathname for every source and destination. If you enter a pathname in either option, you must also explicitly specify the public directory if you want the `uucico` daemon to be allowed to place files in that location.

Additional NOREAD/NOWRITE locations

These options allow you to explicitly specify directories and files on the local system to which the remote system cannot transfer data. These are exceptions to the READ and WRITE options.

Commands

A list of commands the remote system is allowed to run on the local system. If you list a set of commands, that list comprises the new default command set for the systems listed in the MACHINE entry of the `/usr/lib/uucp/Permissions` file. The default is the command `rmail` only.

VALIDATE option

If you want the calling remote system to use a specific ID and password, check Yes; otherwise, check No.

If you use this option, no other ID from the remote system can call in. Several systems, however, can use the same ID. The VALIDATE option is meaningful only when the login ID and password are protected.

CALLBACK option

If you want the local system to contact the remote system before the remote system can transfer any files to the local system, check Yes; otherwise, check No.

If both systems use the CALLBACK option in their respective `Permissions` files, they will never be able to communicate with each other.

Phone number (for modem)

For modem connections, the phone number and speed of the modem attached to the local system. You must provide this information to the administrator of each remote system that will connect to your system as an incoming system.

5.3 Configuring UUCP

After you complete the required UUCP planning, use the `uucpsetup` script to configure UUCP. To invoke the `uucpsetup` script, enter the following command:

```
# /usr/sbin/uucpsetup
```

By default, the `uucpsetup` script prompts you for the information required to configure connections, incoming systems, and outgoing systems. To configure only specific components, you can specify one of the other options listed in Table 5–1.

Table 5–1: Options for uucpsetup Command

Use this command:	If you want to:
`uucpsetup`	Configure connections, incoming systems, and outgoing systems
`uucpsetup –i`	Configure the incoming systems only
`uucpsetup –o`	Configure the outgoing systems only
`uucpsetup –p`	Configure the `Poll` file

The following sections provide information on how to configure connections, incoming systems, outgoing systems, and the `Poll` file.

5.3.1 Configuring Connections

After you invoke `uucpsetup`, use the the information you gathered in Section 5.2.2.1 to configure UUCP connections. The following guidelines explain how to answer some of the script questions:

- Device names — The script lists the available device names. Enter the last letter or number of the device that you want the modem to use. For example, if you want to use tty01, enter 1.

- `/etc/inittab` entry ID — The script prompts you for the *Identifier* field and asks if this entry will be used in shared mode. It automatically supplies information for the other fields. No two processes can have the same ID.

  The following example illustrates how to select the process ID (PID) u4:

  ```
  Select an ID for the process in /etc/inittab file
  For example type 'u1': u4
  ```

  The ID that you select is checked against those that exist in the `/etc/inittab` file. If the ID that you assign exists, the `uucpsetup` script prompts you to enter another ID.

  You must also indicate whether the system will use the modem or direct line in shared mode.

  For more information on the `/etc/inittab` file, see `inittab`(4).

5.3.2 Configuring Outgoing Systems

After you invoke the `uucpsetup` script, use the the information you gathered in Section 5.2.2.2 to configure UUCP for outgoing systems. This enables you to use UUCP to connect to other remote systems.

If you are doing a complete UUCP setup, the `uucpsetup` script prompts you for information on outgoing systems when you finish configuring connections. The following guidelines explain how to answer some of the script questions:

- Phone number — If you choose a dialing prefix and the telephone number, the script prompts you to enter a prefix to be defined in the `/usr/lib/uucp/Dialcodes` file. After you enter the prefix, the script prompts you for the meaning of the prefix. Enter the sequence of numbers that you want the system to substitute for the prefix. The following example illustrates how to define the prefix btown to be the dialing sequence 1617772:

  ```
  Enter the prefix for the Dialcodes file; for example "boston"
         stands for 9=16171234 : btown
  What telephone number does the prefix stand for; Please include
         the long distance access code, area, or country codes;
         for example type 9=1617123 : 9=1617772
  ```

  The 9 in this example is used to obtain a secondary dial tone. The 9 is site specific; it can be different for your site. The equal sign (=) is used with the 9, or number for your site, and means "wait for the dial tone." Following the equal sign (=) is the rest of the number.

- Password — For security considerations, the password is not written on the worksheet. However, when the script prompts for it, you must enter it.

If you define an outgoing TCP system, edit the /etc/uucp/Systems file and add an entry for the remote system. The remote system name must be the fully qualified name.

When you finish configuring your outgoing system, you need to configure the /usr/lib/uucp/Poll file. See Section 5.3.4 for more information.

5.3.3 Configuring Incoming Systems

After you invoke the uucpsetup script, use the the information you gathered in Section 5.2.2.3 to configure UUCP for incoming systems. This enables specific remote systems to connect to your system using UUCP.

If you are doing a complete UUCP setup, the script prompts you for information on incoming systems when you are done configuring outgoing systems.

The first time you add an incoming system, the Incoming Systems Configuration menu prompts you for the name of the system you want to add. After you add an incoming system, this menu offers you the following choices:

- Specify a remote system name.
- Specify options for all the other systems not specified in the Permissions file but listed in the Systems file.
- Neither. If you choose this option, the script terminates and the defaults for the options are not entered in the Permissions file.

The following guidelines explain how to answer some of the script questions:

- Password — The uucpsetup script invokes the vipw command, which starts your default editor (defined in the EDITOR environment variable) and allows you to edit the UUCP entry for the incoming system. After you are finished editing the /etc/passwd file, save the file, exit the editor, and supply a password for the new entry. The following example shows output from this process on a system that is configured to use the vi utility as its default editor:

```
Invoking 'vipw'.
Press RETURN to continue...
Return
root:fQPPWjF20Dfso:0:1:Charles Root:/:/bin/csh
nobody:*Nologin:4294967294:4294967294:anonymous NFS user:/:
daemon:*:1:1:Mr Background,,,:/:
uucp:No Login:2:2:UNIX-to-UNIX Copy:/usr/spool/uucppublic:\
     /usr/lib/uucp/uucico
bin:*:3:4:Mr Binary:/bin:
marcy:5jW0VXKeP6n1E:1242:15:Marcy Darcy,,,:\
     /usr/users/marcy:/bin/false
Umachine1:H/kj951Fq12ub:2:2:uucp login:/usr/spool/uucppublic:\
```

```
    /usr/lib/uucp/uucico
~
~
~
"/etc/ptmp" 15 lines, 933 characters
:wq
15 password entries, maximum length 100

You must enter a password
Changing password for Umachine1.
New password:
Retype new password:
```

You must provide this information to the administrator of each remote system that will connect to your system as an incoming system.

- Commands — The script prompts you for each command separately.

If you define an incoming UUCP system and your system uses NIS, edit the `/etc/passwd` file and add the wildcard (`+:`) as the last line (if it is not there already).

5.3.4 Configuring the Poll File

After you finish configuring an outgoing system, you need to configure the `/usr/lib/uucp/Poll` file to schedule the intervals at which the local system will poll remote systems. You can configure the `Poll` file by invoking the `uucpsetup` script with the –`p` option and completing the following steps:

1. Enter 1 (Configure the Poll file) from the Poll File Configuration Menu.
2. Enter the name of the remote system, which has been configured in the `/usr/lib/uucp/Systems` file as an outgoing system.
3. Enter the sequence of hourly intervals. For example, to have the system polled every 4 hours, enter 0 4 8 12 16 20.

   Press Return to update the `Poll` file.
4. To add another system to the `Poll` file, enter y; otherwise, press Return to exit `uucpsetup`.

See `Poll`(4) for more information about the `Poll` file.

5.3.5 Configuring the uucico Daemon

The `uucico` daemon transfers UUCP command, data, and execute files to remote systems. Both the local and remote systems run the `uucico` daemon, and the two daemons communicate with each other to complete transfer requests.

Typically, the `uucico` daemon is set up as the UUCP user's login shell for incoming connections, or it is automatically called by various UUCP commands for outgoing connections, and no further configuration is necessary. However, you might need to specify the type of flow control the `uucico` daemon uses for certain UUCP transfers. For example, if you establish a connection to a terminal server via a modem and then use the `telnet` utility to connect to a UUCP account, you might require a different type of flow control than a user who initiates UUCP transfers via a serial port connection.

To specify the type of flow control that the `uucico` daemon uses, set the `FLWCTL` environment variable for the accounts on your system that use UUCP connections. Permitted values for `FLWCTL` are: HW (hardware), SW (software), HSW (hardware and software), and NONE. The local and remote systems must use the same type of flow control. If the remote site runs UUCP on a different platform, set `FLWCTL` to NONE on the Tru64 UNIX system.

For example, to establish a UUCP connection in a `telnet` session, you would set flow control to NONE as follows:

```
$ export FLWCTL=NONE
$ /usr/lib/uucp/uutry remote_site
```

On a system that is configured to allow other sites to dial in, you can use the following procedure to create a customized script that automatically sets the `FLWCTL` variable:

1. Create a file, optionally called *uu_start*, that contains the following commands:

   ```
   #! /bin/ksh
   export FLWCTL=NONE
   exec /usr/lib/uucp/uucico $*
   ```

2. Change the permissions on the file to make it executable:

   ```
   # chmod +x /usr/local/bin/uu_start
   ```

3. Change the UUCP account's login shell from `/usr/lib/uucp/uucico` to the new executable file:

   ```
   # chsh uucp
   Old shell: /usr/lib/uucp/uucico
   New shell: /usr/local/bin/uu_start
   ```

5.4 Managing UUCP

This section describes how to perform the following UUCP tasks:

- Monitor the file transfer queue

- Clean up the spooling directories
- View the log files
- Clean up the `sulog` and `cron/log` files
- Limit the number of remote executions
- Schedule work in the spooling directory
- Call file transfer programs
- Poll remote systems

5.4.1 Monitoring the File Transfer Queue

Monitoring the file transfer queue enables you to determine the status of several types of networking operations, including jobs that have been queued on a local system for transfer to a remote system. General users and system administrators can monitor the file transfer queue.

5.4.1.1 Getting Queue Status Manually

To get queue status manually, use the `uustat –q` command.

This command lists the jobs queued for all systems. The jobs listed in the queue include jobs that are currently executing as well as jobs that are waiting to execute. If a status file exists for a system, its date, time, and status information are reported.

The `uustat` command also allows you to do the following:

- Get information about the status of mail activities
- Control `uucp` jobs queued to run on remote systems
- Check the status of `uucp` connections to other systems, using the `–m` option
- Cancel transfer requests, using the `–k` option
- Monitor requests for file transfers generated with the `uucp` and `uuto` commands, and requests for command executions generated with the `uux` command

See `uustat`(1) for more information on `uustat` options.

The following example shows all jobs in the current queue: one command file for system host4, three command files for system host6, and two command files for system host8. The command files for system host6 have been in the queue for two days.

```
# uustat –q
host4 1C Sat May 9 11:12:30 1992 SUCCESSFUL
host6 3C(2) Sat May 9 11:02:35 1992 CAN'T ACCESS DEVICE
```

```
host8 2C Sat May 9 10:54:02 1992 NO DEVICES AVAILABLE
```

5.4.1.2 Getting Queue Status Automatically

You can automatically receive status information about the uucp file transfer queue. To enable this mechanism, edit the /usr/spool/cron/crontabs/uucp file and delete the comment character (#) from the beginning of the following line:

```
# 48 8,12,16 * * * /usr/lib/uucp/uudemon.admin > /dev/null
```

In the preceding example:

48	Represents minutes
8,12,16	Represents hours based on 24-hour clock notation
* * *	Three asterisks are placeholders representing the day of the month, the month of the year, and the day of the week

The cron daemon will run the uudemon.admin shell script daily at 48 minutes past the hours 8, 12, and 16; that is, at 8:48 a.m., 12:48 p.m., and 4:48 p.m. The uudemon.admin script sends mail to the uucp login ID containing queue status information.

Note

These times are the defaults. You can change the time to fit the needs of your site by editing the line in the /usr/spool/cron/crontabs/uucp file.

You can also manually run the uudemon.admin script by entering the following command:

```
# /usr/lib/uucp/uudemon.admin
```

5.4.1.3 Guidelines for Checking Queue Status

When examining queue status, check the number and age of the file-transfer and command execution requests queued in the /usr/spool/uucp/*system_name* directory. In some cases, queued jobs remain in the queue for some time, essentially going undelivered. The status information you need to check includes:

- The age in days of the oldest request in each queue
- The number of times the local system has tried and failed to reach the specified computer
- The reason for the failure to contact the specified system

See Appendix D for error messages and solutions.

If necessary, delete the files in the queue, either manually or automatically. See Section 5.4.2 for information on deleting files.

5.4.2 Cleaning Up the Spooling Directories

Each system connected by UUCP has the following spooling directories:

- The `/usr/spool/uucp/`*`system_name`* directory is the UUCP spooling directory. It contains queued local requests for file transfers and command executions on remote systems. These files are removed by the `uucp` program after they are transferred to the designated system.
- The `/usr/spool/uucppublic` directory is the UUCP public directory. When a user transfers a file to a remote system or issues a request to execute a command on an other system, the files generated by these UUCP commands are stored in the public directory on the designated system.

Depending upon the size of your installation and the number of files sent to the local `/usr/spool/uucppublic` directory by users on remote systems, the public directory can become quite large. Similarly, if requests are not transferred to remote systems for whatever reasons, the spooling directory could also become quite large. Therefore, part of UUCP management is to clean up the spooling directories and conserve disk resources.

5.4.2.1 Cleaning Up Directories Manually

To clean up the spooling directories manually, log in as root and remove the files by using the `uucleanup` command.

The `uucleanup` program performs the following tasks:

- Informs the system manager of requests to send files to and receive files from remote systems that the local system cannot contact.
- Warns users about requests that have been waiting in the spooling directory for a given period of time. The default is 1 day.
- Returns to the original sender mail that cannot be delivered.
- Removes all other files older than a specified number of days from the spooling directory.

Note

Depending on the size of your installation and the available storage space on the local system, you can set the age limit for any length of time. However, it is best to allow files to

remain in the spooling directory for at least the default number of days.

See uucleanup(8) for more information on the uucleanup command options.

The following example deletes all old files in the UUCP spooling and public directories for system host2 on the local system:

```
# uucleanup –shost2
```

5.4.2.2 Cleaning Up Directories Automatically

Although automatic cleanup is not enabled when UUCP is installed, you can enable it by doing the following:

1. Log in as root.
2. Edit the /usr/spool/cron/crontabs/uucp file and delete the comment character (#) from the beginning of the following line:

   ```
   # 45 23 * * * ulimit 5000; /usr/lib/uucp/uudemon.cleanu > /dev/null
   ```

   In the preceding example:

45	Represents minutes
23	Represents hours based on 24-hour clock notation
* * *	Three asterisks are placeholders representing the day of the month, the month of the year, and the day of the week

   The cron daemon will start the uudemon.cleanu shell script daily at 45 minutes after hour 23; that is, at 11:45 p.m. The shell script in turn starts the uucleanup program. This time is the default. You can change the time to fit the needs of your site by editing the line in the /usr/spool/cron/crontabs/uucp file.

You can instruct the cron daemon to run the uudemon.cleanu shell script daily, weekly, or at longer intervals, depending on the number of uucico and uuxqt transactions that occur on the local system.

The uudemon.cleanu script incorporates the actions of the uucleanup program and performs the following additional tasks:

- Locates and deletes empty directories and files older than 30 days from the /usr/spool/uucppublic directory. This helps keep the local file system from overflowing when users send files to the public directory. If the local system does not have enough storage space to accommodate a large /usr/spool/uucppublic directory, you can change the 30-day default to a shorter time period by modifying the uudemon.cleanu shell script.

- Cleans up all the uucp spooling directories, including the public directories, unless you direct it to clean up only the directories of a specific system by issuing the uucleanup –s *system_name* command.
- Updates archived log files, removing log information more than two days old. The script removes log files for individual computers from the /usr/spool/uucp/.Log directory, merges them, and places them in the /usr/spool/uucp/.Old directory, which contains old log information.
- Mails a summary of the status information gathered during the current day to the UUCP login ID. You can modify the script to send status information to other login IDs, such as root.

The operating system allots UUCP a specified amount of storage space for any one log file; the number of blocks is determined by the default ulimit value. If the uudemon.cleanu script fails to execute because the ulimit value is set too low for the requirements of the local system, increase the default ulimit value.

See uudemon(4) for more information on command options.

5.4.2.3 Guidelines for Removing Files

When removing files from the queue, observe the guidelines for the following files:

- Execute files — Usually, you can remove execute files that have been in the queue for at least two days, using either the uucleanup or uudemon.cleanu script. The execute files are still queued because the data files required to execute the specified command on the designated system were not transferred. Since data files are generally sent at the same time as execute files, the transfer probably failed at the point of destination. Execute files are named X.*filename* and data files are named D.*filename*.
- Command files — Before removing old command files, make every possible effort to establish the connection and transfer the files. You can then remove these files by using either the uucleanup or uudemon.cleanu script. Command files are named C.*filename*.

5.4.3 Viewing Log Files

The uucp program creates a log file for each remote system with which your local system communicates. Each time you use the networking utilities facility, uucp places status information about each transaction in the appropriate log file. Log file names can be in either of the following forms:

/usr/spool/uucp/.Log/*daemon_name/system_name*

/usr/spool/uucp/.Log/*command_name/system_name*

In the preceding example:

daemon_name	Represents either uucico (called by the uucp and uuto commands) or uuxqt (called by the uux command)
command_name	Represents either uucp or uux
system_name	Represents the name of the system with which your local system is communicating

To display individual log files, use the uulog command.

You can use the uulog command to display a summary of uucp and uux requests by user or by system. See uulog(1) for more information on the uulog command and its options.

Instead of viewing the log files individually, you can have the uudemon.cleanu script automatically append these log files to one primary log file, and then view only the primary log file.

The uudemon.cleanu script combines the uucico, uuxqt, uux, and uucp log files on a system and stores them in a directory named /usr/spool/uucp/.Old. By default, the uudemon.cleanu script saves log files that are up to two days old.

You can change the default by modifying the –o2 option in the following line in the uudemon.cleanu script:

```
uucleanup –D7 –C7 –X2 –o2 –W1
```

If storage space is a problem on a particular system, consider reducing the number of days that the files are kept in the individual log files. See Section 5.4.2.2 for information on setting up the uudemon.cleanu script.

The following command displays the log file for uucico requests for system host2:

```
# uulog –s host2
```

The following command displays the log file for uuxqt requests for system host1:

```
# uulog –x host1
```

The following command displays the last 40 lines of the file transfer log for system host6 and executes a tail –f command. Press Ctrl/c to terminate the command.

```
# uulog –f host6 –40
```

5.4.4 Cleaning Up sulog and cron/log Files

The following two system log files are affected by the uucp program:

- The /usr/adm/sulog file contains a history of superuser (su) command usage. The uudemon entries in the /usr/spool/cron/crontabs/uucp file each use the su command.
- The /usr/adm/cron/log file contains a history of all the processes generated by the cron daemon.

Both files can grow quite large over a period of time. Purge these files periodically to keep them at a reasonable size. See *System Administration* for information on these files.

5.4.5 Limiting the Number of Remote Executions

The Maxuuxqts file, located in the /usr/lib/uucp directory, limits the number of uuxqt processes running simultaneously on a local system. Typically, the file requires no configuration or maintenance unless the system on which it is installed is utilized frequently and heavily by users on remote systems.

To change the number of uuxqt processes on the system, edit the Maxuuxqts file and change the ASCII number to meet the needs of your installation; the default is 2. In general, the larger the number, the greater the potential load on the local system.

5.4.6 Scheduling Work in the Spooling Directory

When users issue uucp commands to copy files and execute remote commands, the files containing these work requests are queued for transfer in the local /usr/spool/uucp/*system_name* directory. The UUCP uusched daemon schedules the transfer of these files.

5.4.6.1 Starting uusched Manually

You can start the uusched daemon manually to schedule jobs by executing the uusched command. See uusched(8) for a list of available options.

5.4.6.2 Starting uusched Automatically

Although you can start the uusched daemon manually, the preferred method is to start it automatically at specified intervals by using the uudemon.hour shell script, which is stored in the /usr/lib/uucp directory. The shell script, in turn, is started periodically by the cron daemon, based on instructions in the /usr/spool/cron/crontabs/uucp file.

The /usr/lib/uucp/Maxuuscheds file limits the number of remote systems that the uucico daemon can contact at any one time. This file is used in conjunction with the uusched daemon and the lock files in the

/usr/spool/locks directory to determine the number of systems currently being polled.

The Maxuuscheds file requires no configuration or maintenance unless the system on which it is installed is utilized frequently and heavily by users on remote systems. You use this file to help manage system resources and load averages.

The Maxuuscheds file contains as number that you can change to meet the needs of your installation; the default is 2. In general, the larger the number, the greater the potential load on the local system.

See uusched(8) for more information on the uusched command and its options.

The following command starts the uusched daemon manually as a background process:

```
# /usr/lib/uucp/uusched &
```

5.4.7 Calling File Transfer Programs (uudemon.hour)

The uudemon.hour shell script is used in conjunction with the Poll file, the uudemon.poll shell script, and the /usr/spool/cron/crontabs/uucp file to initiate calls to remote systems. Specifically, uudemon.hour calls programs involved in transferring files between systems at specified hourly intervals.

You can instruct the cron daemon to run the uudemon.hour shell script at specified hourly intervals. The frequency at which you run the script depends on the amount of file transfer activity originating from the local computer.

Although the uudemon.hour shell script is not enabled when UUCP is installed, you can enable it by doing the following:

1. Log in as root.
2. Edit the /usr/spool/cron/crontabs/uucp file and delete the comment character (#) from the beginning of the following line:

   ```
   # 25,55 * * * * /usr/lib/uucp/uudemon.hour > /dev/null
   ```

   In the preceding example:

25,55	Represents minutes past the hour
* * * *	Four asterisks are placeholders representing the hour interval, the day of the month, the month of the year, and the day of the week

The cron daemon will run the uudemon.hour script at 25 minutes past the hour and again at 55 minutes past the hour; for example, at 8:25 a.m. and 8:55 a.m., 9:25 a.m. and 9:55 a.m., and so on.

These times are the defaults. You can change the time to fit the needs of your site by editing the line in the /usr/spool/cron/crontabs/uucp file.

If users on the local system initiate a large number of file transfers, you might need to specify that the cron daemon run the uudemon.hour script several times an hour. If the number of file transfers originating from the local system is low, you can probably specify a start time once every 4 hours, for example.

5.4.8 Polling Remote Systems (uudemon.poll)

The uudemon.poll shell script is used in conjunction with the Poll file, the uudemon.hour shell script, and the /usr/spool/cron/crontabs/uucp file to initiate calls to remote systems. The uudemon.poll shell script polls the systems listed in the /usr/lib/uucp/Poll file. In addition, it creates command files for the systems listed in the Poll file.

The time at which you run the uudemon.poll script depends on the time at which you run the uudemon.hour script. You generally schedule the polling shell script to run before the hourly script. This schedule enables the uudemon.poll script to create any required command files before the cron daemon runs the uudemon.hour script.

Although the uudemon.poll script is not enabled when UUCP is installed, you can enable it by doing the following:

1. Log in as root.
2. Edit the /usr/spool/cron/crontabs/uucp file and delete the comment character (#) from the beginning of the following line:

   ```
   # 20,50 * * * * /usr/lib/uucp/uudemon.poll > /dev/null
   ```

   In the preceding example:

20,50	Represents minutes past the hour
* * * *	Four asterisks are placeholders representing the hour interval, the day of the month, the month of the year, and the day of the week

   The cron daemon will run the uudemon.poll script at 20 minutes past the hour and again at 50 minutes past the hour, for example, at 8:20 a.m. and 8:50 a.m., 9:20 a.m. and 9:50 a.m., and so on.

   These times are the defaults. You can change the times at which the cron daemon executes the uudemon.poll script to correspond to the

times you set up for the `uudemon.hour` script. Set the `cron` daemon to run the `uudemon.poll` script approximately 5 to 10 minutes before running the `uudemon.hour` script.

6

Network Time Protocol

The Network Time Protocol (NTP) provides accurate, dependable, and synchronized time for hosts on both wide area networks (WANs) like the Internet network and local area networks (LANs). In particular, NTP provides synchronization traceable to clocks of high absolute accuracy, and avoids synchronization to clocks keeping bad time. The Tru64 UNIX NTP subsystem is derived from the University of Delaware's implementation, NTP Version 4.0.98a.

This chapter describes:

- The Tru64 UNIX NTP subsystem and its components (Section 6.1)
- How to plan for your NTP configuration (Section 6.2)
- How to configure your system to use NTP (Section 6.3)
- How to enable the high-resolution clock (Section 6.4)
- How to monitor hosts that are running NTP (Section 6.5)
- How to query servers that are running NTP (Section 6.6)

For introductory information on NTP, see `ntp_intro`(7). For troubleshooting information, see Section 9.11. Also, for information about the latest releases of NTP, more examples of how to configure NTP subnets, and more extensive NTP troubleshooting information, visit the NTP website at **http://www.eecis.udel.edu/~ntp**.

As an alternative to NTP, you can set your system time by using the `rdate` command or the `timed` daemon.

Note

The `timed` daemon is provided only for compatibility; use NTP for time synchronization. If you plan to run both the `timed` daemon and NTP, configure NTP first and run the `timed` daemon with the `–E` option.

For more information on the `rdate` command, see `rdate`(8) and `ntp_manual_setup`(7).

For more information on the `timed` daemon, see `timed`(8) and `timedsetup`(8).

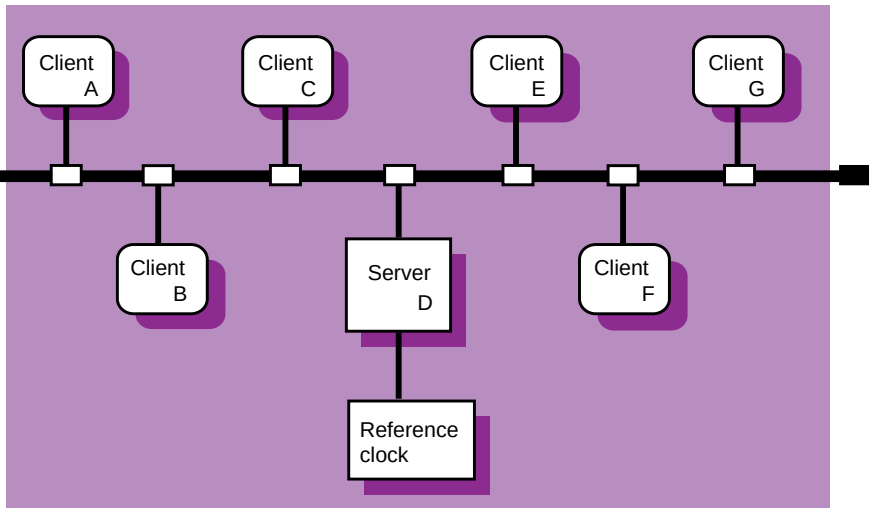
6.1 NTP Environment

In the NTP environment, systems can have the following roles:

- Client — An NTP client system is a system that synchronizes its time with local NTP servers.
- Server — An NTP server is a local system that synchronizes its time with an Internet NTP server or with a local reference clock, or both for better accuracy.

Figure 6–1 shows a sample NTP configuration on a LAN in which host D is an NTP server that uses a local reference clock as its time source. Hosts A, B, C, E, F, and G are NTP clients, synchronizing their time with host D.

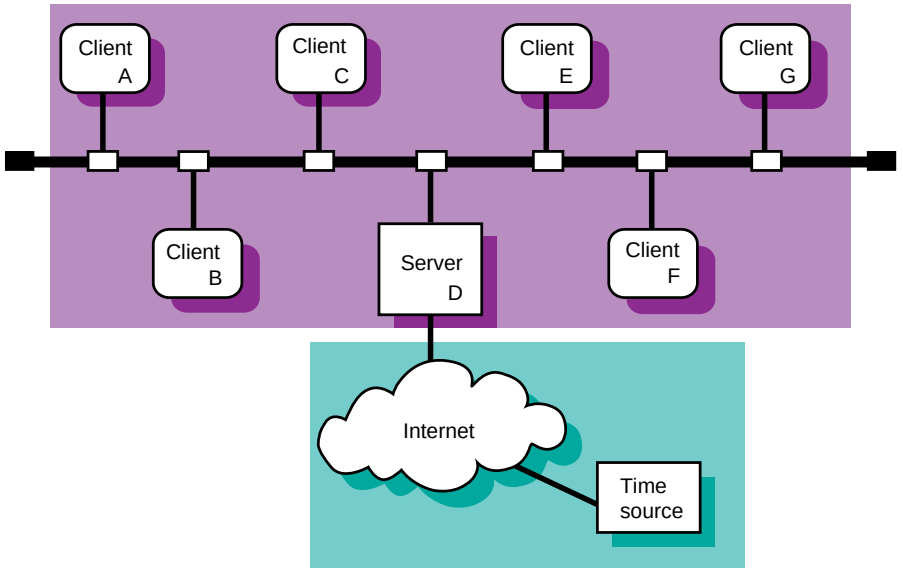
Figure 6–1: Sample NTP Configuration (Local Clock)



ZK-1158U-AI

Figure 6–2 shows a sample NTP configuration in which host D is an NTP server that uses an Internet time server as its time source. Hosts A, B, C, E, F, and G are NTP clients, synchronizing their time with host D.

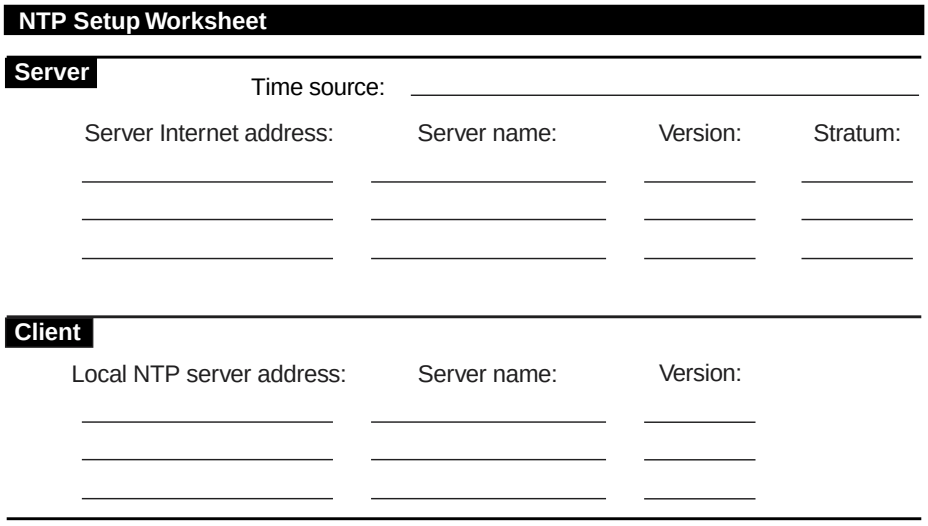
Figure 6–2: Sample NTP Configuration (Internet Source)



6.2 Planning NTP

Your system can be a local NTP server or an NTP client, or both. Figure 6–3 shows the NTP Setup Worksheet, which you can use to record the information required to configure NTP. If you are viewing this manual online, you can use the print feature to print a copy of this worksheet. The following sections explain the information you need to record on the worksheet.

Figure 6–3: NTP Setup Worksheet



NTP Setup Worksheet

Server

Time source: _____

Server Internet address:	Server name:	Version:	Stratum:
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____

Client

Local NTP server address:	Server name:	Version:
_____	_____	_____
_____	_____	_____
_____	_____	_____

6.2.1 Server Information

Time source

Your system's time source. For local NTP servers, the time source is one of the following:

- Internet NTP servers — If your system is connected to the Internet, you can obtain a list of possible NTP Internet servers from **http://www.eecis.udel.edu/~ntp** on the World Wide Web. Select a minimum of three systems from the server list with which to synchronize the time on your local NTP servers. Obtain permission from the contact person listed for each Internet server before specifying it as a server for your local NTP servers.
- A reference clock — If your network is not connected to the Internet network, you can select a system on your network to configure with a reference clock, which obtains its time via radio broadcasts or satellite transmissions. As a last resort, if no Internet servers or reference clock devices are available, you can select a system on your network to configure with a local reference clock, which means that the system uses its own CPU timekeeping unit as a reference clock.

  See `ntp_manual_setup`(7) and `ntp.conf`(4) for information about configuring different types of reference clocks.

Server Internet address

The IP address of the Internet NTP server or the local reference clock. Local NTP servers are the time sources for NTP clients.

Server name

The host name of the Internet NTP server.

Version

The version of NTP daemon running on the Internet NTP server or the local reference clock. This can be Version 1 (the `ntpd` daemon), Version 2 (the `xntpd` daemon), or Version 3 (the `xntpd` daemon). Servers running Version 3.2 or earlier of the Tru64 UNIX operating system run Version 2 (the `xntpd` daemon); servers running Version 4.0 or later of the Tru64 UNIX operating system run Version 3 (the `xntpd` daemon).

Stratum

A stratum value describes the accuracy of a system's reference clock: the higher the number, the less accurate the clock.

If you are configuring a local reference clock, you can specify a higher stratum value to indicate that the clock's time is not very accurate. This discourages other systems from using your clock as a reliable time source, because NTP clients will obtain the time from the server with the lowest stratum they can find. For example, if you set a stratum of 8 for your local reference clock, NTP clients will ignore your server and use a server with stratum 2 or lower (if one can be found).

You can supply a value from 0 to 15 for the Stratum field; however, it is best not to override the default value assigned by NTP unless you have a specific reason for doing so. For local reference clocks, that default value is 3. For other clocks, the default value is 0.

6.2.2 Client Information

Local NTP server address

The local NTP server IP address. Local NTP servers are the time sources for NTP clients.

Server name

The local NTP server name.

Version

The version of NTP daemon running on the local NTP server. This can be Version 1 (the `ntpd` daemon), Version 2 (the `xntpd` daemon),

or Version 3 the (the xntpd daemon). Servers running Version 3.2 or earlier of the Tru64 UNIX operating system run Version 2 (the xntpd daemon); servers running Version 4.0 or later of the Tru64 UNIX operating system run Version 3 (the xntpd daemon).

6.3 Configuring NTP

Use the SysMan Menu application of the Common Desktop Environment (CDE) Application Manager to configure NTP servers and clients. To invoke the SysMan Menu application, follow the instructions in Section 1.2.1.

Note

You might need to manually edit your /etc/ntp.conf file if you require a more complex NTP configuration than the SysMan Menu can produce; for example, if your NTP server uses a local or external reference clock as a time source. If you manually edit the ntp.conf file, do not use the SysMan Menu to modify the configuration in the future. The SysMan Menu recognizes only a small subset of the options that you can use in the ntp.conf file, and might overwrite your configuration.

See ntp_manual_setup(7) and ntp.conf(4) for more information about manually configuring NTP.

Also, if you plan to use both NTP and the timed daemon, set up NTP prior to setting up the timed daemon.

To configure NTP, do the following:

1. From the SysMan Menu, select Networking→Additional Network Services→Network Time Protocol (NTP)→Configure system as an NTP client to display the Configure NTP Client dialog box.

   Alternatively, enter the following command on a command line:

   ```
   # /usr/bin/sysman ntp_config
   ```

2. Indicate whether you want to enable authentication by selecting the appropriate check box. If you choose to enable authentication, you must enter at least one authentication key as follows; repeat the steps to add additional keys:

   a. Select Add under the Authentication Keys list to display the Add/Modify dialog box.

   b. Enter the Key Number and Key for a peer or peers. The Key Number is a number from 1–15 that identifies the Key. The Key is an alphanumeric password of 1–8 characters with no spaces.

c. Select OK to add the authentication key to the list and to dismiss the Add/Modify dialog box.

Your authentication keys are stored in the `/etc/ntp.keys` file when you save your configuration and close the Configure NTP Client dialog box.

3. Select Add under the Servers & Peers list to display the Add/Modify dialog box.
4. Enter the host name, mode, version, and key number for an NTP server. If the NTP Server's IP address is not available through DNS or NIS, you must add it to the `/etc/hosts` database on your system as described in *Network Administration: Connections*.

   For clients, enter the information for an NTP server that is local to your site.

   For servers, enter the information for an Internet NTP server or a local reference clock. (See Section 6.2 for information.) If you are configuring a local reference clock and you need to override the default stratum that the `xntpd` daemon assigns to it, select the Fudge Factor check box and select a value from 0 to 15 for the Stratum field.

   The information will be recorded in the `/etc/ntp.conf` file. For clients, entries in this file are designated as server entries because clients can synchronize their time only with these systems. An NTP server, however, can contain server and peer entries in its `ntp.conf` file. A peer system can be synchronized to another system's time or it can synchronize another system's time to its own.
5. Select OK to validate the parameters you entered and to dismiss the Add/Modify dialog box. To add other NTP servers, repeat steps 3 through 5. It is best to specify at least three servers.
6. Indicate whether you want to correct large time differences by selecting the appropriate check box.

   This option, enabled by default, allows `xntpd` to correct differences of more than 1000 seconds between your system time and your system's NTP server's time that occur after the `xntpd` daemon is started. The `ntpdate` command is run at boot time by the `/sbin/init.d/settime` script to correct initial time differences. If your system is sensitive to security threats or the clock that you have chosen as your server is not stable, do not enable this option. If you do not use this option, time differences of more than 1000 seconds will cause the `xntpd` daemon to log a message to the `syslogd` daemon and exit.
7. Indicate whether you want to prevent time from being set backwards by selecting the appropriate check box. The default is to allow the `xntpd` daemon to set the system time backward.

8. Select OK to accept the configuration and to close the Configure NTP Client dialog box.
9. A new dialog box is displayed indicating that the changes have been saved and prompting you to start the `xntpd` daemon.
10. Select Yes to start the daemon and apply your changes immediately, or select No to close the Configure NTP Client dialog box and apply the changes the next time you reboot your system.

Note

When you start NTP, the system attempts to synchronize its clock with an NTP server's clock. If you previously enabled a screen saver on your system, the time difference might be enough to activate it. In some cases, this blanks the screen, but it does not harm the system. Move the mouse or hit a key on the keyboard to reactivate the display.

If you choose Yes, you are informed that the NTP daemons have been started. Select OK to dismiss the message and to close the Configure NTP Client dialog box.

You can modify your NTP configuration after the initial setup. You can also stop and restart the `xntpd` daemon as necessary. See the online help for more information.

6.4 Enabling the High-Resolution Clock

The operating system includes an optional high-resolution clock that can be used for time-stamping and for measuring events that occur on the order of microseconds, such as the time spent in a critical code path. Programmers might be able to use this information to find the source of a bug or to determine where a program can be optimized to improve performance.

To enable the high-resolution clock, add the following line to the kernel configuration file and rebuild the kernel:

```
options MICRO_TIME
```

The system clock (`CLOCK_REALTIME`) resolution as returned by the `clock_getres` function does not change, nor does the timer resolution. However, the time as returned by the `clock_gettime` routine is extrapolated between the clock ticks, and the granularity of the time returned is in microseconds. The resulting time values are SMP-safe, they are monotonically increasing, and they have an apparent resolution of 1 microsecond.

6.5 Monitoring Hosts Running the xntpd Daemon

You can monitor the hosts running the `xntpd` daemon by using either the `ntpq` command or the `xntpdc` command.

To monitor the local host's NTP status using the `ntpq` command, use the following syntax:

ntpq [*options...*]

To monitor remote hosts' NTP status using the `ntpq` command, use the following syntax:

ntpq [*options...*] *host1 host2...*

Table 6–1 shows the `ntpq` command options.

Table 6–1: Options to the ntpq Command

Option	Function
–c *subcommand*	Interprets *subcommand* as an interactive format command and adds it to a list of commands to be executed on the specified host or hosts
–i	Forces `ntpq` to operate in interactive mode
–p	Prints a list of peers and a summary of their state

You can specify `ntpq` subcommands on the command line with the `-c` option, or you can run the `ntpq` program interactively with the `-i` option. When you are finished entering subcommands in interactive mode, enter `quit` to exit the program.

By default, the subcommands apply to the local host. You can specify a host other than the local host on the command line or with the `host` subcommand in interactive mode. See `ntpq`(8) for more information about this command and its subcommands.

The following example shows normal output from the `ntpq` command with the `-p` option (or `peers` subcommand):

```
% ntpq –p
     remote           refid      st  when poll reach  delay  offset   disp
==============================================================================
*host2.corp.com  host121.corp.co  2    47   64  377    31.3   93.94   16.5
+host4.corp.com  host2.corp.com   3   212 1024  377    33.8   89.58   16.9
 host8.corp.com  host2.corp.com  16 never   64    0     0.0    0.00  64000
```

The last line of the previous example shows that host8 is either not running NTP or cannot be reached.

To monitor the local host's NTP status using the `xntpdc` command, use the following syntax:

xntpdc [*options...*]

To monitor remote hosts' NTP status using the xntpdc command, use the following syntax:

xntpdc [*options...*] *host1 host2...*

Note

The latest versions of the xntpdc command and xntpd daemon, delivered with NTP Version 4, are incompatible with previous versions of NTP. If you use the latest xntpdc command to collect information from an older xntpd daemon, or an older xntpdc command to collect information from the latest xntpd daemon, you will receive inconsistent results.

Table 6–2 shows some of the xntpdc command options.

Table 6–2: Options to the xntpdc Command

Option	Function
–c *subcommand*	Interprets *subcommand* as an interactive format command and adds it to a list of commands to be executed on the specified host or hosts.
–i	Forces xntpdc to operate in interactive mode.
–l	Prints a list of peers that are known to the server.
–p	Prints a list of peers and a summary of their state. This is similar in format to the ntpq –p command.

See xntpdc(8) for more information on this command and its subcommands.

The following example shows normal output from the xntpdc command with the –p option:

```
% xntpdc –p
     remote           refid      st  when poll reach  delay  offset   disp
==============================================================================
*host2.corp.com host121.corp.co   2    47   64   377   31.3   93.94   16.5
+host4.corp.com host2.corp.com    3   212 1024   377   33.8   89.58   16.9
.host5.corp.com host12.usc.edu    2   111 1024   377   39.1   46.98   17.7
```

6.6 Querying Servers Running NTP

You can query time by using the ntp and ntpdate commands. However, it is best to use the ntpdate command because it works with all versions of NTP and provides additional features.

7

Mail System

The Tru64 UNIX mail system enables users to send mail to other users, whether on the same system, same network, or the other side of the world. This chapter describes:

- The Tru64 UNIX mail system and its components (Section 7.1)
- How to plan for your mail configuration (Section 7.2)
- How to configure mail (the `sendmail` utility) on a standalone system or across an enterprise (Section 7.3)
- How to configure Post Office Protocol (POP) mail (Section 7.4)
- How to configure Internet Message Access Protocol (IMAP) mail (Section 7.5)
- How to administer mail on server and client systems (Section 7.6)
- Utilities you can use to process and receive mail (Section 7.7)

For additional introductory information on mail, see `mail_intro`(7), the *sendmail* book by O'Reilly & Associates, and the *Sendmail Installation and Operation Guide* (provided in PDF format on the Tru64 UNIX Documentation CD-ROM). For troubleshooting information, see Section 9.12 for the `sendmail` utility and Section 9.13 for POP and IMAP mail.

The mail daemons in Tru64 UNIX are based on `sendmail` Version 8.11.1 from Sendmail, Inc, POP3 Version 3.0.2 from Qualcomm, Inc., and Cyrus IMAP4 Version 1.6.24 from Carnegie-Mellon University. If you need later versions of these packages than the operating system offers, you can obtain updated software directly from the aforementioned organizations or you can obtain the Internet Express for Tru64 UNIX product (formerly Open Source Internet Solutions), a collection of popular Open Source software that HP distributes on a CD-ROM.

The Internet Express kit usually contains more up-to-date versions of the Open Source software than the operating system because the kit is updated and distributed several times a year. Internet Express also contains an administration utility that allows you to easily configure advanced features of `sendmail`, including masquerading, virtual domains, anti-spam, and the Lightweight Directory Access Protocol (LDAP). For more information about the Internet Express product, see the following URL:

http://tru64unix.compaq.com/internet/prod_sol.htm

7.1 Mail Environment

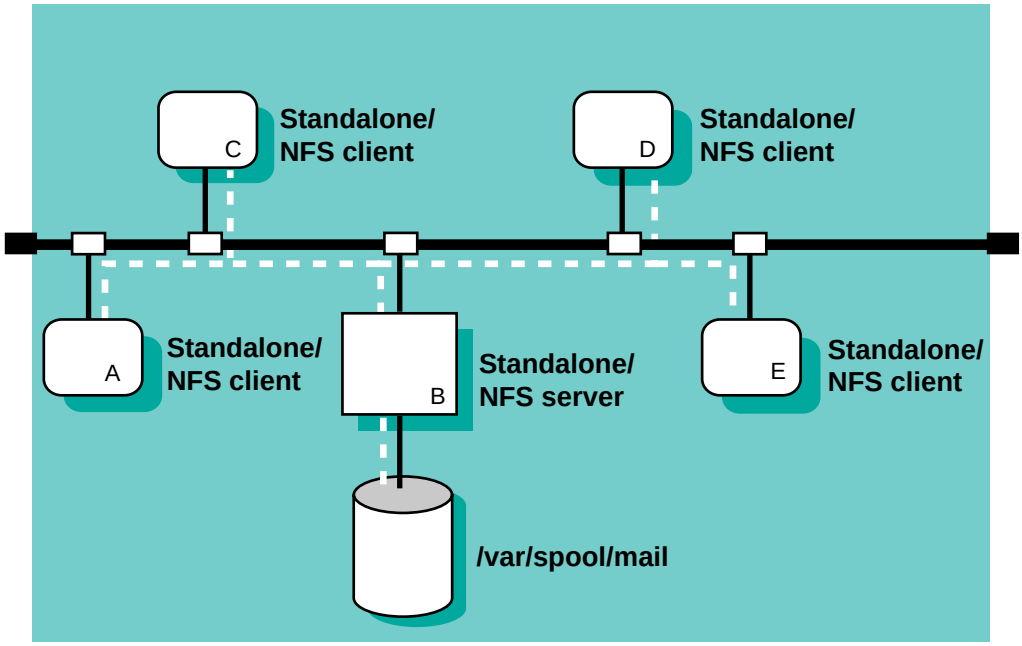
In the mail environment, systems can have the following roles:

- Standalone — A mail standalone system is one that processes, sends, and delivers mail locally. This is useful for configurations of from 1 to 6 systems. In small LAN configurations of two or more systems, one system serves the mailbox to the other systems using NFS. In this case, NFS must be configured on all systems.
- Client — A mail client system is a system that sends all of its mail to a mail server for processing and delivery. If the addressee is on the client system, the mail is delivered there. If not, the mail is forwarded to the destination system.
- Server — A mail server system is a system that receives mail from clients in a local domain for processing and delivery to other domains, the Internet, or other networks. In addition, the server also receives mail from other domains for delivery.

Figure 7–1 shows a sample standalone configuration on a LAN in which all hosts are configured as mail standalone systems. Host B is also an NFS server, exporting the `/var/spool/mail` directory to hosts A, C, D, and E. Hosts A, C, D, and E are also NFS clients, importing the `/var/spool/mail` directory from host B.

The hosts must also have identical information in their `passwd` and `aliases` files. This information can be distributed either by using NIS or by manually editing the files on each system.

Figure 7–1: Sample Mail Standalone Configuration



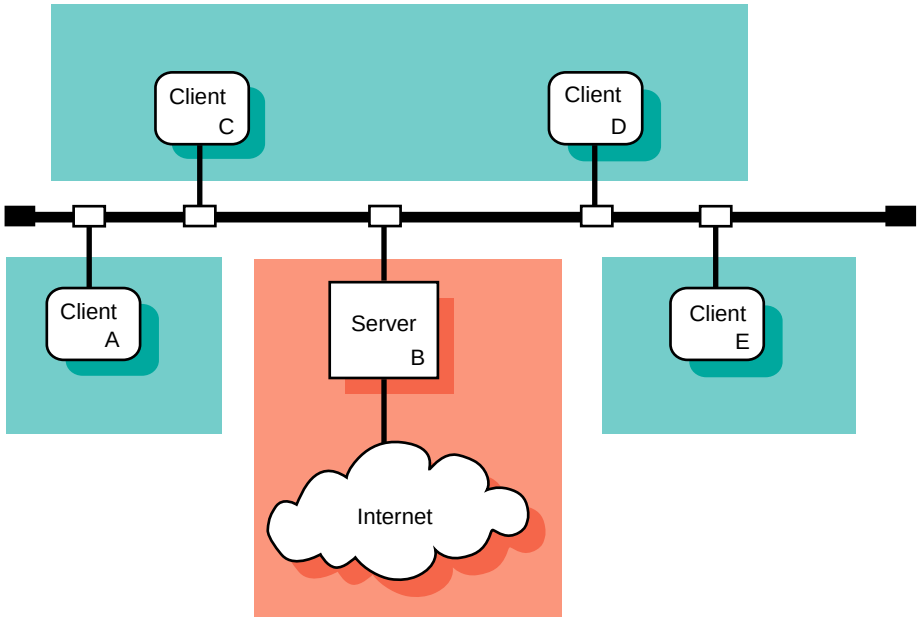
ZK-1156U-AI

Figure 7–2 shows a sample client/server configuration in which host B is configured as a mail server and hosts A, C, D, and E are configured as mail clients. This is useful in larger enterprise networks that consist of multiple domains and connections to the Internet or other networks.

This configuration also provides for the creation of a natural hierarachy of mail servers in large enterprise networks with multiple domains. Mail clients in each domain direct all traffic to one or more mail servers, depending on the number of clients in the domain. Each domain's servers then forward mail to the enterprise's top domain servers for forwarding to the Internet. Since almost all of your local domain's mail traffic goes through the servers, this simplifies administration and problem resolution in that you have to manage only the servers.

The connection to the Internet in Figure 7–2 can be direct or through a local access provider. Business configurations typically use firewalls and dedicated mail servers. If using a firewall, ensure the firewall and the mail server are configured to work with each other. See the documentation for your firewall product for more information.

Figure 7–2: Sample Mail Client/Server Configuration



ZK-1157U-AI

If users need to send mail between systems that use different mail protocols, such as the Simple Mail Transfer Protocol (SMTP), UNIX-to-UNIX Copy Program (UUCP), and DECnet, it is best to designate specific server systems in your network to perform those functions. These server systems are also known as mail relays.

Additional mail configurations are possible, but they require more effort to plan for and to configure. See the *sendmail* book by O'Reilly and Associates and the *Sendmail Installation and Operation Guide* for more information.

In implementing a client/server mail environment, you need to decide how to do the following:

- Direct outgoing mail to the servers
- Handle incoming mail to the domain
- Deliver mail to clients
- Distribute the `aliases` file
- Distribute the `passwd` file
- Handle DECnet mail

This section describes each of these topics.

7.1.1 Directing Outgoing Mail to Servers

To direct outgoing mail to a server, you include the DNS mail exchanger (MX) entry in the `/etc/namedb/hosts.db` file. This entry specifies a system in the local domain that can deliver mail to other systems, especially those not directly connected to the local network. Using MX to route mail has the following benefits:

- You can define an MX record to point to all of the mail servers in your local domain. If a mail server is inaccessible, mail can be delivered to another host listed in the MX record.
- You can use MX records to define a system to be a mail exchanger for an inaccessible remote system. Then, if you send mail to the remote, inaccessible host, instead of being queued on your local system and periodically resent, the mail is sent to the mail exchanger and queued there until the host is restored.

For information on adding entries to the `/etc/namedb/hosts.db` file, see Section 2.8.2, Appendix G, and `bind_manual_setup`(7).

7.1.2 Handling Incoming Mail to the Domain

To simplify the handling of incoming mail to a domain and to ensure reliability, use domain-based addresses in your environment. Mail sent over the Internet is usually addressed in the following format:

username@hostname.domain

For example: `joe@host1.nyc.big.com`

Using domain-based addresses, this address appears as follows:

`joe@nyc.big.com`

Mail is sent to the local domain `nyc.big.com` instead of to the specific host within that domain `host1.nyc.big.com`; the return address is also `@nyc.big.com`. Then, the mail servers within the local domain decide how to deliver the mail to the user's account.

Domain-based addresses make it easier to manage your mail environment. You can change your mail system (that is, move user accounts and replace or move systems) without disrupting your mail delivery. These changes are transparent to users sending mail to your systems.

7.1.3 Delivering Mail to Clients

Once mail is delivered to the domain, you can deliver it to clients using one of the following mechanisms:

- Deliver the mail to the `/var/spool/mail` directory on each client, which is the default
- Deliver the mail to the server and use NFS to serve the mail directory to each client
- Deliver the mail from a server to a local client machine using POP (see Section 7.4)
- Deliver the mail to a server using IMAP (see Section 7.5)

To deliver mail to each client, each server in the domain must have an aliases file that contains an entry for each user on the client. For example:

```
username1: username1@client1
username2: username2@client1
```

7.1.4 Distributing the aliases File

For standalone and server systems, use the Network Information System (NIS) to distribute the mail aliases file from one machine. In a LAN environment with standalone systems, distribute the mail aliases file from the NFS server system. In a client/server environment, distribute the `aliases` file to the servers in the domain. In any case, sharing the `aliases` file among systems simplifies administration in that you need to update only one aliases file, instead of several.

See `aliases`(4) for more information about the database. See Section 7.6.3 and Chapter 3 for information about distributing the database with NIS.

7.1.5 Distributing the passwd File

If you have multiple server systems in a domain, make sure that the information in the `passwd` file is identical on each system. For security reasons and to ensure correct mail delivery, it is best to do this by manually editing the `passwd` file on each server system.

7.1.6 Handling DECnet Mail

When you set up a mail server system, you must consider that the mail address formats for DECnet Phase IV and DECnet/OSI are different from those for TCP/IP. Therefore, you need to establish a mapping scheme to translate mail addresses when sending mail between a DECnet node and a TCP/IP node.

The mapping scheme used by the Tru64 UNIX version of the `sendmail` program for DECnet Phase IV encapsulates DECnet addresses inside a pseudodomain. For example, a typical DECnet Phase IV address has the following format:

nodename::*username*

Mail addressed in this format is mapped to an address in the following format:

username@ *nodename.pseudodomain.top.domain*

The variables represent the following:

username

The user name.

nodename

The DECnet node name.

pseudodomain

An arbitrary string that specifies the DECnet pseudodomain. The pseudodomain can be an arbitrary string, but it must be used consistently throughout your organization. All of your mail systems must be configured to use the same string for the pseudodomain.

top.domain

Usually, your company's domain name; for example, `abc.com`.

The mapping for DECnet/OSI uses a similar scheme. A typical DECnet/OSI address has the following format:

username@ *namespace*:.*site.nodename*

Mail addressed in this format is mapped as follows:

username@ *nodename.site.namespace.pseudodomain.top.domain*

As with DECnet Phase IV, the pseudodomain can be an arbitrary string. However, if you use both DECnet Phase IV and DECnet/OSI within your organization, it is best to use different pseudodomain names.

Some environments that support both DECnet Phase IV and DECnet/OSI use the DECnet Phase IV syntax to handle DECnet-based mail. This simplifies the mail administration task. In order to implement this, all DECnet-OSI nodes must have a unique Phase IV Synonym and must be configured to use the Phase IV Synonym. You can reconfigure a DECnet/OSI host by typing the following command line:

```
# ncl set session control application mail11 Node Synonym=true
```

See the DECnet/OSI documentation for more information.

7.2 Planning Mail

This section describes those tasks you need to do before configuring mail.

7.2.1 Verifying That Required Protocols Are Installed

Depending on the protocols supported by your mail server, verify that the following required subsets are installed and configured:

- DECnet
- DECnet/OSI
- X.25 (PSInet)
- UUCP

See the documentation for each product for installation and configuration instructions. For UUCP, verify that the UUCP subset is installed by entering the following command:

```
# setld –i | grep OSFUUCP
```

If it is not installed, install it by using the `setld` command. For more information on installing subsets, see `setld`(8) or the *Installation Guide*.

7.2.2 Verifying That Required Services Are Configured

The following table lists specific mail configurations and the network service required:

If you want to:	Configure this service:
Distribute the aliases file	NIS
Use domain-based addressing	DNS/BIND

If NIS is needed, enter the following command as root to verify that NIS is configured:

```
# rcmgr get NIS_CONF
```

If the command returns `NO`, then NIS is not configured. See Chapter 3 for instructions on how to configure NIS and distribute the `aliases` file.

If DNS is needed, enter the following command as root to verify that DNS is configured:

```
# rcmgr get BIND_SERVERTYPE
```

If the command returns nothing, then DNS is not configured. See Chapter 2 for instructions on how to configure DNS.

7.2.3 Preparing for the Configuration

After you install and configure the required protocols and services, you configure mail using the Mail Configuration application.

Mail configuration consists of:

- Defining the standalone, client, or server system
- Defining the protocol information (server systems only)

The following sections contain worksheets that you can use to record the information required to configure mail.

7.2.3.1 General System Information

Figure 7–3 shows the Basic Mail Setup Worksheet. If you are viewing this manual online, you can use the print feature to print a copy of this worksheet. The following sections explain the information you need to record on the worksheet.

Figure 7–3: Basic Mail Setup Worksheet

Basic Mail Setup Worksheet

Mail server (clients only): _____

Top domain (servers only): _____

Mailbox directory: ☐ Local ☐ NFS client ☐ NFS server

Locking: ☐ lockf ☐ Lock file ☐ Both

Mailbox server: _____

Mail server (clients only)

The fully qualified name of your mail server; for example, `foo.dec.com`. Or, the name of your domain; for example, `dec.com`, if you are using domain-based routing. It is advantageous to specify the domain name itself because your mail service cannot be interrupted by a single mail server that becomes unavailable.

Top domain (servers only)

The name of the highest level domain in your organization that uniquely identifies your organization. For example, if the server domain name is `nyc.big.com`, the top domain is `big.com`. If the server domain name is `cs.big.univ.ac.uk`, the top domain is `big.univ.ac.uk`.

Mailbox directory

The location of the mailbox directory.

For standalone and client systems, if the mailbox directory is on the local system, check Local. If it is on a remote system and is to be mounted on the local system using NFS, check NFS Client. If the local system is to export mail boxes to NFS clients, check NFS Server.

For server systems, check Server to make the mailbox directories available to other systems. If you do not want to share the mailbox directories, check Local. In this case, use the `aliases` file to send each user's mail to the appropriate system. See Section 7.6.3 and `aliases`(4) for more information.

Locking

The type of file locking to use on the mailbox.

For standalone and client systems, if the host with the mailbox directory is a Tru64 UNIX system, check lockf; this provides the best performance. If you are not sure what operating system the host with the mailbox directory is running, check Lock file. If you want to use both, check Both.

Note

The locking mechanism you select must match the mechanism used by the NFS server. If you are not sure how the locking mechanisms are set on the NFS server, ask the administrator of the NFS server.

For server systems, if you checked Local as the mailbox location, check lockf. If you checked Client as the mailbox location, check Lock file. If you checked Server as the mailbox location, check Both.

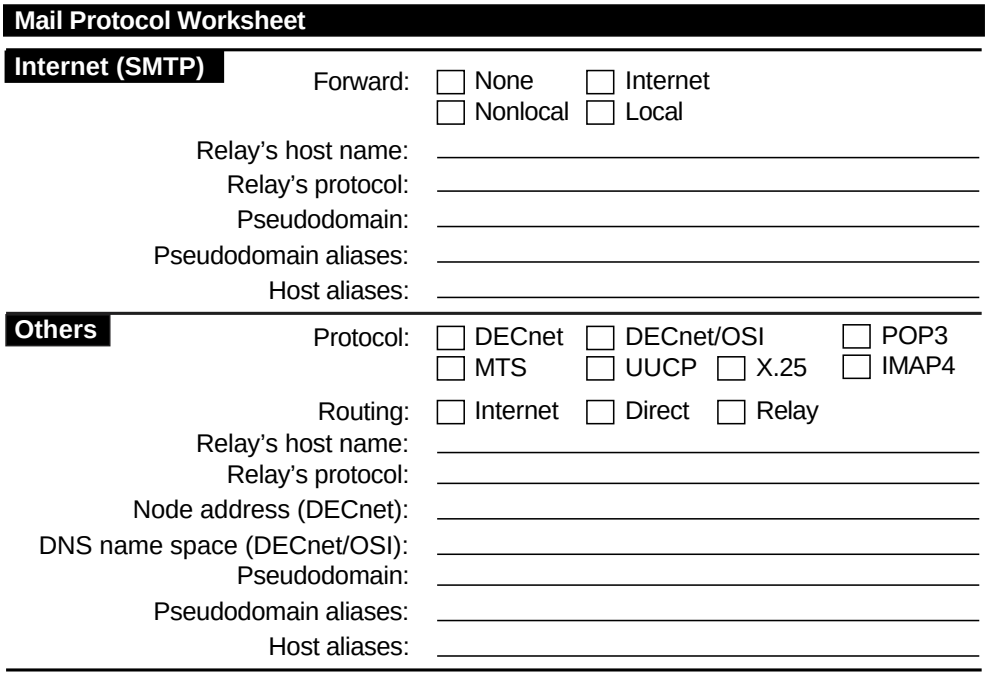
Mailbox server

The name of the system that exports the mailbox to your local system.

7.2.3.2 Protocol Information

Figure 7–4 shows the Mail Protocol Worksheet. If you are viewing this manual online, you can use the print feature to print a copy of this worksheet. The following sections explain the information you need to record on the worksheet.

Figure 7–4: Mail Protocol Worksheet



Mail Protocol Worksheet

Internet (SMTP)

Forward: ☐ None ☐ Internet ☐ Nonlocal ☐ Local

Relay's host name: _____

Relay's protocol: _____

Pseudodomain: _____

Pseudodomain aliases: _____

Host aliases: _____

Others

Protocol: ☐ DECnet ☐ DECnet/OSI ☐ POP3 ☐ MTS ☐ UUCP ☐ X.25 ☐ IMAP4

Routing: ☐ Internet ☐ Direct ☐ Relay

Relay's host name: _____

Relay's protocol: _____

Node address (DECnet): _____

DNS name space (DECnet/OSI): _____

Pseudodomain: _____

Pseudodomain aliases: _____

Host aliases: _____

To configure your system as an Internet (SMTP) server, you need to collect the following information:

Forward

The type of mail that must be forwarded to a relay. If the local host has direct access to the Internet and does not forward any mail, check None. If the local host must forward all mail addressed outside of the top domain, check Internet. If the local host must forward all messages addressed outside of the local Internet domain, check Nonlocal. If the local host must forward all mail, including local domain mail, check Local.

Relay's host name

The name of the remote host that will process SMTP mail.

Relay's protocol

The name of the protocol the server uses to forward messages to the relay host.

Pseudodomain

An arbitrary string that specifies the pseudodomain for SMTP mail. The pseudodomain name must be unique for each protocol and must be used consistently throughout your enterprise.

Pseudodomain aliases

Any synonyms for your pseudodomain.

Host aliases

The alternative names that other systems might use to direct mail to your host.

To configure your system as a server for other mail protocols, you need to collect the following information:

Protocol

The type of mail protocols to use. Available protocols include the following:

- DECnet (Phase IV)
- DECnet/OSI (Phase V)
- Internet Mail Protocol (SMTP) (required)
- Internet Message Access Protocol (IMAP)
- Message Transport System (MTS)
- Post Office Protocol (POP)
- UUCP
- X.25 (PSInet)

Routing

For DECnet, DECnet/OSI, UUCP, MTS, and X.25 only. If mail for the particular protocol is to be forwarded over the Internet to an unspecified gateway, check Internet. The Internet depends on DNS to select an appropriate relay; therefore, do not specify a relay hostname for the Internet.

If the particular protocol is installed on this server, check Direct. If mail requiring the particular protocol is to be forwarded to another system for processing, check Relay. Complete the Relay's hostname and Relay's protocol fields.

Relay's host name

The name of the remote host that will process mail for the protocol.

Relay's protocol

The name of the protocol the server uses to forward messages to the relay host.

Node address (DECnet)

The address for this machine (DECnet only).

DNS name space (DECnet/OSI)

The complete DNS name space name for this node (DECnet/OSI only). The syntax of the DNS name space is as follows:

namespace:.site.nodename

Pseudodomain

An arbitrary string that specifies the pseudodomain (DECnet, DECnet/OSI, and MTS only). The pseudodomain name must be unique for each protocol and must be used consistently throughout your enterprise.

Pseudodomain aliases

Any synonyms for your pseudodomain (DECnet, DECnet/OSI, UUCP, and MTS only).

Host aliases

The alternative names that other systems might use to direct mail to your host.

7.3 Configuring Mail

Use the Mail Configuration application of the Common Desktop Environment (CDE) Application Manager to configure mail on systems with graphics capabilities.

Note

Alternatively, you can use the SysMan Menu (`/usr/sbin/sysman mailsetup`) or the `mailsetup` utility to configure mail on

your system. See the online help and `mailsetup`(8) for more information.

You can configure the following systems:

- Standalone systems
- Client systems
- Server systems

To start the Mail Configuration application, do the following:

1. Log in as root.
2. Click on the Application Manager icon on the CDE desktop.
3. Double-click on the System_Admin application group icon.
4. Double-click on the Configuration application group icon.
5. Double-click on the Mail Configuration application icon in the Configuration group. The Mail Configuration main window is displayed, showing available Mail service types and configured Mail service types.

To exit the Mail Configuration application, choose File then Exit. See `mailconfig`(8) for more information.

The Mail Configuration application has an extensive online help system that you can use, instead of the instructions in this section, to configure mail on your system.

7.3.1 Configuring a Standalone Mail System

To configure mail for a standalone system, do the following:

1. Select Standalone from the Available Mail Service Types list box in the Mail Configuration window
2. Select Configure to display the Standalone Setup dialog box.
3. Select Mailbox Setup to display the Mailbox Setup dialog box if your site uses NFS to import or export system mailbox directories (for instance, `/var/spool/mail`); otherwise, go to step 7, the default settings are applicable for your mail configuration.
4. If your system imports its mailbox using NFS, select the NFS Client radio button and do the following:

   a. Enter the server name in the Mailbox Server field.

   b. Select a radio button for the appropriate Locking mechanism: lockf, Lock Files, or Both.

5. If your system distributes mailboxes to NFS clients, select the NFS Server radio button, then select the Both radio button for the Locking Mechanism setting.
6. Select OK to complete the mailbox setup and close the Mailbox Setup dialog box.
7. Select Commit to save the changes. You are asked if you would like to restart the `sendmail` daemon.
8. Select Restart to start the `sendmail` daemon and apply your changes immediately. Or, select No to apply the changes the next time you reboot your system.

   If you choose Restart, you are informed that the `sendmail` daemon has been started. Select OK to dismiss the message.
9. Select Close to close the Standalone Setup dialog box.

7.3.2 Configuring a Mail Client

To configure a mail client, do the following:

1. Select Client from the Available Mail Service Types list box in the Mail Configuration window.
2. Select Configure to display the Client Setup dialog box.
3. Enter the name of a mail server for outgoing mail in the Mail Server field.
4. Select Mailbox Setup to display the Mailbox Setup dialog box.
5. Select the NFS Client radio button for the Mailbox Directory if your site uses NFS to share system mailbox directories; otherwise, select Local and go to step 7.
6. Enter the name of the server that exports the mailbox directory to your system in the Mailbox Server field.
7. Select a radio button for the appropriate Locking mechanism: lockf, Lock Files, or Both.
8. Select OK to complete the mailbox setup and to close the Mailbox Setup dialog box.
9. Select Commit to save the changes. You are asked if you would like to restart the `sendmail` daemon.
10. Select Restart to start the `sendmail` daemon and apply your changes immediately. Or, select No to apply the changes the next time you reboot your system.

    If you choose Restart, you are informed that the `sendmail` daemon has been started. Select OK to dismiss the message.

11. Select Close to close the Client Setup dialog box.

7.3.3 Configuring a Mail Server

To configure a mail server, follow these steps. If you intend to implement the POP or IMAP daemons, configure SMTP and other necessary protocols first, then see Section 7.4 and Section 7.5.

1. Select Server from the Available Mail Service Types list box in the Mail Configuration window.
2. Select Configure to display the Server Setup dialog box.
3. Select the mail protocol you want to configure from the Available Protocols list box. The Internet Mail Protocol (SMTP) protocol is the only required protocol configuration. Configure additional protocols as necessary.
4. Select Configure to display the protocol setup dialog box for the protocol you selected.
5. For the SMTP protocol, select the type of forwarding for this server. If you select None, go to step 11; otherwise, go to step 7.
6. For the DECnet, DECnet/OSI, MTS, UUCP, and X.25 protocols, select a Routing type. If you select Internet or Direct, go to step 9. If you select Relay, go to step 7.
7. Enter a host name in the Relay's Hostname field if you will be forwarding mail to another system for processing; otherwise, continue with step 9.
8. Select the protocol used to communicate with the relay in the Relay's Protocol pull-down menu.
9. For the DECnet, DECnet/OSI, and MTS protocols, in the Pseudo Domain field, enter the domain name used to identify mail that requires the selected protocol.
10. For the DECnet, DECnet/OSI, MTS, UUCP, and X.25 protocols, to add aliases for the pseudodomain, select Pseudo Domain Aliases to display the Pseudo Domain Aliases dialog box, and do the following:

    a. Enter the alias name in the Alias field and select Add.

    b. Repeat the previous step as many times as necessary.

    c. Select OK to close the Pseudo Domain Aliases dialog box.

11. To add aliases for this mail server, select Host Alias to display the Host Aliases dialog box, and do the following:

    a. Enter the alias name in the Alias field and select Add.

b. Repeat the previous step as many times as necessary.

c. Select OK to close the Host Aliases dialog box.

12. For the DECnet protocol, enter the DECnet node address (area.node) for this server in the Node Address field, for example, `32.958`.
13. For the DECnet/OSI protocol, enter the name space of the node, which is usually the token before the colon (:) in a DECnet Phase V address, in the DNS Name Space field.
14. Select OK to close the Setup dialog box for the protocol you selected. The Server Setup dialog box is active.
15. Configure another protocol if necessary. Repeat steps 3 through 15 for each additional protocol.
16. Select Mailbox Setup to display the Mailbox Setup dialog box.
17. Select a radio button for Mailbox Directory.

    If your site does not use NFS to distribute the system mailbox directories, select Local instead of NFS Server, and then go to step 19.
18. If you selected NFS Client as a Mailbox Directory, enter the name of the mail server in the Mail Server field. Be sure to include the domain. For example, for a server named `mailhub`, the server name with domain might be `mailhub.nyc.dec.com`.
19. Select a radio button for the appropriate Locking mechanism: lockf, Lock Files, or Both.
20. Select OK to complete the mailbox setup and close the Mailbox Setup dialog box.
21. Select Commit to save the changes. You are prompted to restart the `sendmail` daemon.
22. Select Restart to start the `sendmail` daemon and apply your changes immediately. Or, select No to apply the changes the next time you reboot your system.

    If you choose Restart, you are informed that the `sendmail` daemon has been started. Select OK to dismiss the message.
23. Select Close to close the Server Setup dialog box.
24. Add DNS mail exchanger (MX) records to the `/etc/namedb/hosts.db` file for each host in your environment, if necessary. See Section 7.1.1 for more information.

7.3.4 Adding a New Mail Host

To add a new mail host to your existing mail environment, do the following:

1. Configure the network and network services on the host. See *Network Administration: Connections* for more information.
2. If you are using DNS MX records in your environment, update the DNS data files. See Section 7.1.1 for more information.

7.4 Post Office Protocol

The Post Office Protocol Version 3 (POP3 or POP) is a client/server protocol that allows users to download their e-mail from a mail server to a remote client. It is intended for users that mainly access their e-mail in an offline mode. In offline mode, messages are delivered to a server and reside there until the user connects to the server and downloads the incoming messages to the client machine (a desktop or laptop computer running Windows, Macintosh, UNIX, or another operating system). Thereafter, all message processing is local to the client machine and environment. This is the mode used widely today by Internet Service Providers (ISPs) to provide e-mail services for their consumers. See `pop3d`(8) for further information.

7.4.1 Installing POP

The operating system provides a POP3 server (`/usr/sbin/pop3d`) from Qualcomm, Incorporated, which is fully installed and configured for you when you install the `OSFINET` subset (check the installation log file for any warnings or errors). The `pop3d` daemon is configured to listen on port 110 for incoming connections, and allows any user of the system to access their e-mail via a POP client.

During installation, the `/etc/passwd`, `/etc/services`, and `/etc/inetd.conf` configuration files are updated. If the lines displayed in the following examples are not present in the configuration files, the POP3 service may not behave appropriately. If a previous version of POP was detected, or if the `OSFINET` subset did not install properly, the files might not have been updated and the changes must be made manually.

The `/etc/passwd` file must contain the following line; if it does not, add the line to the file:

```
pop:*:13:6:POP Mail Service Account:/:
```

If necessary, change the user identification number, 13, to a value that is appropriate for your system.

The `/etc/services` file must contain the following line; if it does not, add the line to the file:

```
pop3    110/tcp
```

The /etc/inetd.conf file must contain the following line; if it does not, add the line to the file:

```
pop3    stream  tcp     nowait  root    /usr/sbin/pop3d    pop3d
```

7.4.2 Migrating to the New POP3 Implementation

The POP service was upgraded in Tru64 UNIX Version 5.0. Migration paths are provided for systems that were running either the version of POP3 offered with the OSFMH (RAND Corp. Mail Handler) subset or the Qualcomm POP3 service (if your version came directly from Qualcomm).

If you use the MH POP3 service, you must migrate your POP user accounts from the /usr/spool/pop/POP file to the mailauth database and convert your mailboxes to the new format.

If you use the Qualcomm POP3 service, you must migrate your POP user accounts from the popauth database to the mailauth database; however, you do not need to convert your mailboxes. The only difference between Qualcomm POP3 and the Tru64 UNIX implementation of Qualcomm POP3 is the mail authorization database, which has been enhanced to store secondary POP and IMAP passwords.

The following sections describe the migration paths for each service. See popcv(8) for further information.

7.4.2.1 Migrating from MH POP3

To migrate from MH POP3 service to the new implementation, complete the following tasks:

1. Remove any startup scripts for the /usr/lib/mh/popd file in the /sbin/rc directories.
2. Make sure that the /etc/inetd.conf and /etc/services configuration files were updated with the correct entries as described in Section 7.4.1.
3. Initialize the mailauth database by entering the following command:

   # **/usr/bin/mailauth -init**

4. Use the popcv utility to move usernames and passwords from the /usr/spool/pop/POP file to the mailauth database (/etc/pop.auth.pag and /etc/pop.auth.dir). Enter the following command, where *filename* can be an alternate file used to store POP passwords:

   # **/usr/bin/popcv [*filename*]**

5. Use the `mailcv` tool to convert existing MH POP3 mail folders to the new POP3 format:

   a. Change directory to the MH POP3 mail folder directory:

      # **cd /usr/spool/mail/POP**

      The directory might be `/usr/spool/pop` or another directory depending on how you configured MH POP3.

   b. For each mail user, enter the following command, where *input* is the file name of the user's MH POP3 folder:

      # **/usr/dt/bin/mailcv -Q -f *input***

      Typically, the file name is the same as the POP user's username. For instance, for a user named Jake, you would convert the `/usr/spool/mail/POP/jake` file.

      Optionally, you can change the name of a mail folder during the conversion process by appending the new file name to the end of the command, as in the following example:

      # **/usr/dt/bin/mailcv -Q -f charlie chuck**

      See `mailcv`(1) for more information.

7.4.2.2 Migrating from Qualcomm POP3

To migrate from Qualcomm's POP3 service to the new implementation, complete the following tasks:

1. Ensure that the `/etc/inetd.conf` and `/etc/services` configuration files were updated with the correct entries as described in Section 7.4.1.
2. If a previous `popauth` database exists, convert it to a `mailauth` database by using the following command:

   # **/usr/bin/mailauth -convert**

You need to convert your mail folders only if you previously ran the MH POP3 server.

7.4.3 Configuring a POP Mail Account

To configure a POP mail account, create a UNIX account for the user (if one does not already exist) as described in the *System Administration* manual. The user's mailbox is set up automatically.

Once the user's account is set up on the server, the user can configure a mail application compatible with POP3; for example, Netscape Communicator, which is bundled with the operating system software. At a minimum, you

must provide the user with the following information about mail service in your facility:

- POP username — Specify if different from the UNIX username.
- POP-specific password — Specify if different from the UNIX password.
- POP server name — The mail application collects incoming mail from this server.
- SMTP server name — The mail application delivers outgoing mail to this server.
- Domain name — The mail application adds this domain name to all unqualified addresses for domain-based mail addressing.

7.4.4 Changing Login Authentication

The POP service typically authenticates user accounts by verifying the supplied user name and password against information in the UNIX password file (usually the `/etc/passwd` file). The Tru64 UNIX implementation of POP has been enhanced to optionally support SIA interfaces for authentication on a C2 secure system.

For increased security, the system administrator can have POP users use alternate passwords instead of their usual login passwords; therefore, if a POP password is compromised across the network, system access is not at risk.

There are two ways to enable alternate passwords for POP authentication:

- Arrange for POP users to store alternate passwords in the `mailauth` database (`/etc/pop.auth.dir` and `/etc/pop.auth.pag`).
- Add mail users to the same `mailauth` database as Authenticated POP (APOP) users. APOP uses an encrypted authentication mechanism, also with alternate passwords, that is more secure than standard POP; however, users need mail client applications compatible with APOP to take advantage of it.

Use the SysMan Menu application of the Common Desktop Environment (CDE) Application Manager to enable either authentication option. To invoke the SysMan Menu application, follow the instructions in Section 1.2.1, then follow these steps:

Note

If users in your environment use one of the old POP3 implementations, you must migrate them to the new POP3

implementation as described in Section 7.4.2 prior to enabling alternate passwords.

---

1. From the SysMan Menu, select Mail→Manage users' mail accounts. The Mail User Administration window is displayed. Optionally, you can invoke this utility by executing the following command:

   ```
   # sysman mailusradm &
   ```

2. Select the radio button for List Specific Users and select Complile List.
3. Enter the username or a wildcard in the dialog box and select OK.
4. Select the name of the user for whom you would like to require an alternate password.
5. Select the desired mail service type from the pull-down menu. To require that the user use an alternate password for POP mail, select POP with Mail Password. To switch the user's mail service to APOP, select APOP with Mail Password.
6. Select OK to save your changes.
7. Enter an alternate password for POP or APOP and select OK.

   The user can later set a new password by issuing the `mailauth` command without any options. For example:

   ```
   % /usr/bin/mailauth
   ```

8. Select OK to dismiss the message that indicates that the account has been modified successfully.
9. Select Exit to close the Mail User Administration window.

If you need to change authentication for multiple accounts, you can select List All Local Mail Users in step 2. Hold down the Ctrl key and click the right mouse button to select more than one user name from the list. See `mailusradm`(8) for more information about the Mail User Administration utility.

Optionally, you can use the `mailauth` utility to set up authentication. See `mailauth`(8).

7.4.5 Administrative Tools

You can use the following tools to administer the POP service:

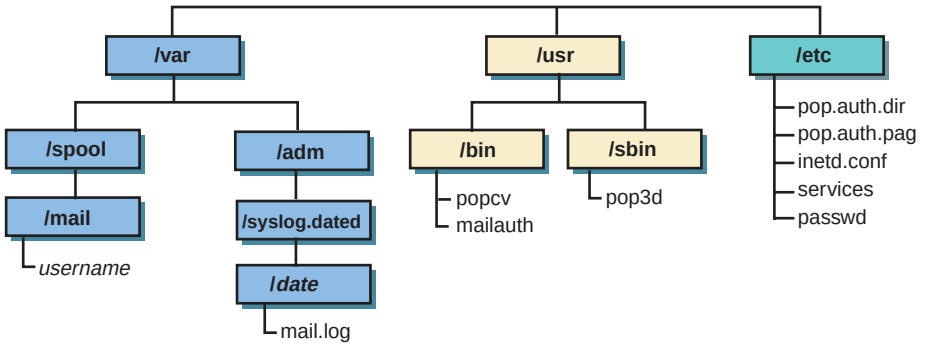
- `mailauth` — Utility used to manage the secondary mail authorization database. See `mailauth`(8).
- `mailusradm` — System administration GUI utility used to configure mail users. See `mailusradm`(8).

The POP server sends log messages to the `syslogd` daemon. The log information is stored in the `/var/adm/syslog.dated/`*`date`*`/mail.log` file. You can use this data to solve problems. The severity levels are NOTICE for failed and successful authentications and DEBUG for all debugging information.

7.4.6 Directory Structure

The POP configuration and mail files are distributed across the file system as indicated in Figure 7–5.

Figure 7–5: POP Directory Structure



ZK-1540U-AI

Table 7–1 describes the purpose of these files and directories.

Table 7–1: POP3 Files and Directories

File or Directory	Purpose
`/etc/passwd` file	Contains account information for each user on the system. Users configured in this file are able to use POP mail by default.
`/etc/pop.auth.*` files	Contain the encrypted mail authorization database, which is used to authenticate POP and IMAP users. See `mailusradm`(8) and `mailauth`(8) for information about editing this database.
`/var/spool/mail` directory	Contains the mail folders for all POP and UNIX mail users on the system. Each folder is a file with a file name that is usually identical to the user's login name.

7.5 Internet Message Access Protocol

The Internet Message Access Protocol Version 4 (IMAP4 or IMAP) is a client/server protocol that allows mail clients to access mail messages on a server. With it, the user can access mail folders and manipulate the contents remotely without having to log in to the server. The protocol allows clients to create, delete, and rename mail folders, to check for new messages and remove old messages, and to selectively retrieve messages for local viewing. In addition, the user can select messages by attributes and parse messages in the RFC 822 and MIME formats.

This protocol can be used in the offline, online, or disconnected mode. The offline mode is the same as that described in Section 7.4. In online mode, messages are manipulated on the server remotely by mail client programs. In disconnected mode, a mail client connects to the mail server, makes a cache copy of selected messages, and then disconnects from the server, later to reconnect and resynchronize with the server. In both online and disconnected access modes, mail is stored on the server, which is often a necessity for people who use different computers at different times to access their messages.

See `imapd`(8), `deliver`(8), and `imapd.conf`(4) for further information.

7.5.1 Installing IMAP

The operating system software includes the Cyrus IMAP4 Revision 1 server (`/usr/sbin/imapd`) by Carnegie Mellon University, which is installed and configured when you install the `OSFINET` subset (check the installation log file for any warnings or errors). The `imapd` daemon is configured to listen on port 143 for incoming connections.

During installation, the `/etc/passwd`, `/etc/services`, and `/etc/inetd.conf` configuration files are updated. If the lines specified in the following examples are not present in the configuration files, the IMAP service might not behave appropriately.

The `/etc/passwd` file must contain the following line; if it does not, add the line to the file:

```
imap:*:14:6:IMAP Mail Service Account:/:
```

If necessary, change the user identification number, 14, to a value that is appropriate for your system.

The `/etc/services` file must contain the following line. If it does not, add the line to the file:

```
imap    143/tcp
```

The /etc/inetd.conf file must contain the following line. If it does not, add the line to the file:

```
imap    stream  tcp    nowait   imap    /usr/sbin/imapd     imapd
```

7.5.2 Upgrading IMAP

Starting with Version 1.6.1 of the Cyrus IMAP4 Revision 1 server, the IMAP files in the quota and user configuration directories, and optionally, the users' mail directories in the IMAP mail spool, are stored in subdirectories a through z, sorted by the first character of each user name. This arrangement reduces the number of entries in a given directory and consequently increases performance and scalability.

If you are running the IMAP server from a previous version of the operating system, and you are upgrading to Tru64 UNIX Version 5.1*x*, you must convert your quota and user configuration directories to the new format. Optionally, you can sort your IMAP mail spool in the same manner by enabling the hashimapspool option in the /etc/imapd.conf file before converting your configuration directories. See imapd.conf(4) for more information.

To convert your directories to the new format, use the dohash utility. See dohash(8) for more information.

7.5.3 Configuring IMAP Mail Accounts

To enable users to receive IMAP mail, you must complete two tasks. First, if the users do not have accounts on the system, you must create them. See the *System Administration* manual and adduser(8) for more information.

Second, you must change the properties of the users' accounts to indicate that their mail is to be processed by the IMAP server. Use the SysMan Menu application of the Common Desktop Environment (CDE) Application Manager to configure the user's mail service type. To invoke the SysMan Menu application, follow the instructions in Section 1.2.1.

To change the user's mail service type, do the following:

1. From the SysMan Menu, select Mail→Manage users' mail accounts. The Mail User Administration window is displayed. Optionally, you can invoke this application by executing the following command:

   ```
   # sysman mailusradm &
   ```

2. Select the radio button for List Specific Users and select Compile List.
3. Enter the username or a wildcard in the dialog box and select OK.

4. Select the name of the user whose mail service type you would like to change from the list.
5. Select the desired mail service type from the pull-down menu. To require that the user use an alternate password for IMAP mail, select IMAP with Mail Password. Otherwise, select IMAP to use the same password.

   If you enable this option, the alternate passwords are stored in the `mailauth` database, which is located in the `/etc/pop.auth.dir` and `/etc/pop.auth.pag` files. See `mailauth`(8) for more information.
6. Select OK to save your changes.
7. Enter the mail administrator's password. In most cases, this password is the same as the root account password. Select OK.
8. Select the privileges to set on the user's mailbox. In most cases, you will select All to allow the user to read, modify, and delete messages in the mailbox. Select OK.

   If you did not select IMAP with Mail Password in step 5, skip to step 10.
9. Enter an alternate password for IMAP and select OK.

   The user can later select a new password by issuing the `mailauth` command without any options. For example:

   ```
   % /usr/bin/mailauth
   ```
10. Select OK to dismiss the message that indicates that the account has been modified successfully.
11. Select Exit to close the Mail User Administration window.

If you need to set up multiple IMAP accounts, you can select List All Local Mail Users in step 2. Hold down the Ctrl key and click the right mouse button to select more than one user name from the list.

See the online help and `mailusradm`(8) for more information.

Once a user's IMAP account is set up on the server side, the user can configure a mail application compatible with IMAP4, for example, Netscape Communicator, which is bundled with the operating system software. At a minimum, you must provide the user with the following information about mail service in your facility:

- IMAP username — Specify if different from the UNIX username.
- IMAP password — Specify if different from the UNIX password.
- IMAP Mailbox location prefix — `user.`*`username`*
- IMAP server name — The mail application collects incoming mail from this server.

- SMTP server name — The mail application delivers outgoing mail to this server.
- Domain name — The mail application adds this domain name to all unqualified addresses for domain-based mail addressing.

7.5.4 Migrating Users from UNIX and POP3 Mail

To convert an existing UNIX or POP3 mail user to IMAP mail, you must first set up the user's IMAP account as described in Section 7.5.3. Then, use the `mailcv` tool to convert the user's mail folder to the IMAP format as follows:

Note

If you are using MH POP3 or a version of Qualcomm POP3 that did not come with the operating system software, follow the instructions in Section 7.4.2 to convert to the new POP3 implementation before converting to IMAP.

1. Change directory to the UNIX/POP3 mail folder directory:

   ```
   # cd /usr/spool/mail
   ```

2. Assume the user's identity by using the `su` command, as follows:

   ```
   # su username
   ```

   You must be the user to convert the user's mail folder to IMAP format with the `mailcv` command.

3. Enter the following command, where *folder* is the file name of the user's mail folder:

   ```
   % /usr/dt/bin/mailcv -I -f folder
   ```

   You need the user's IMAP password to use this command.

   The mail folder file name is usually the same as the user's username. For instance, for a user named Jake, you would convert the `jake` file.

   Optionally, you can move the converted messages to an IMAP subfolder during the conversion process by appending a subfolder name to the end of the command, as in the following example:

   ```
   # /usr/dt/bin/mailcv -I -f charlie business
   ```

   IMAP subfolders are described in Section 7.5.7. See `mailcv`(1) for more information about the `mailcv` command.

4. Exit the `su` session to the user's account, as follows:

   ```
   % exit
   #
   ```

Mail received after the account is changed to IMAP but prior to the conversion process is not lost. The newly-converted messages are appended to the existing messages in the user's mailbox.

Once a user's UNIX or POP account is converted to an IMAP account on the server, the user must reconfigure the mail application. Ensure that the user has an mail application compatible with IMAP4, for example, Netscape Communicator, which is bundled with the operating system software.

You also need to provide the user with information about mail service in your facility, as specified in Section 7.5.3.

7.5.5 Administrative Tools

You can use the following tools to administer the IMAP server:

- `cyradm` — Command line utility used for configuring and managing users, folders, subfolders, and so on. See `cyradm`(1).
- `deliver` — Utility used to deliver mail to an IMAP mailbox. See `deliver`(8).
- `dohash` — Utility used to convert the IMAP configuration directories from the format for older versions of the Cyrus IMAP4 Revision 1 server to the new format for Version 1.6.1 or higher. Another utility, `undohash`, reverses the process. See `dohash`(8).
- `imapquota` — Utility used to report and fix IMAP mail quota usage. See `imapquota`(8).
- `mailauth` — Utility used to manage the secondary mail password database. See `mailauth`(8).
- `mailusradm` — System administration GUI utility used to configure mail users. See `mailusradm`(8).
- `reconstruct` — Utility used to rebuild IMAP mailboxes. See `reconstruct`(8) for further information.

The IMAP server software sends log messages to the `syslogd` daemon. The log information is stored in the `/var/adm/syslog.dated/`*`date`*`/mail.log` file. You can use this data to solve problems. The severity levels are as follows:

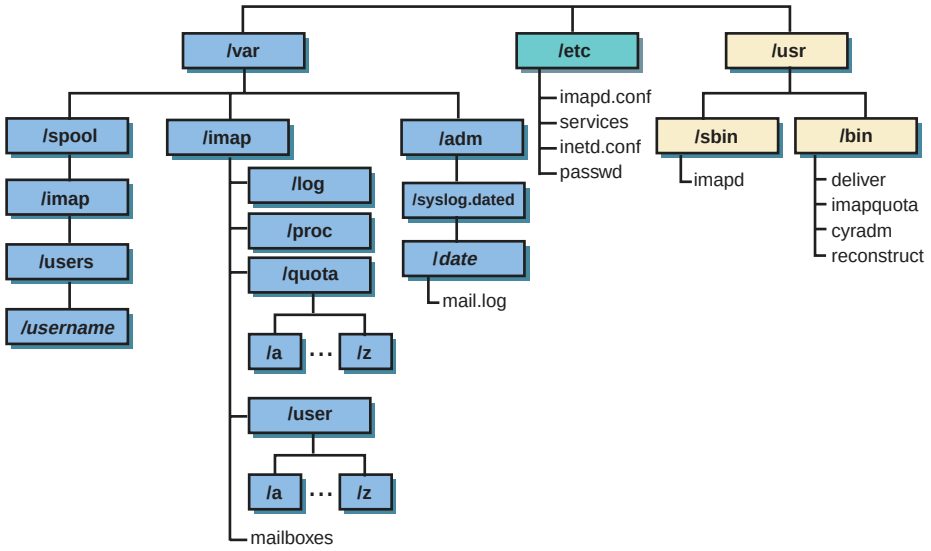
NOTICE	Authentications, both successful and unsuccessful.
ERR	I/O errors, including failure to update quota usage. The message includes the specific file and UNIX error.

WARNING	Protection mechanism failures, client inactivity timeouts.
INFO	Mailbox openings.

7.5.6 Directory Structure

The IMAP configuration and mail files are distributed across the file system as indicated in Figure 7–6.

Figure 7–6: IMAP Directory Structure



ZK-1541U-AI

All of the runtime configuration information is stored in the `/etc/imapd.conf` file. This file contains site configuration and policy options, such as:

- Location of the configuration directory
- Partition names and their corresponding directory roots
- Threshold for quota warning messages
- Whether or not to allow anonymous logins
- Whether or not to automatically create `INBOX` mailboxes for users

See `imapd.conf`(4) for further information.

The configuration directory specified in the `/etc/imapd.conf` file contains the items listed in Table 7–2.

Table 7–2: Configuration Directory Contents

File or Directory	Purpose
`mailboxes` file	Contains a sorted list of each IMAP mailbox on the server along with mailboxes quota root and access control list (ACL), described in Section 7.5.9 and Section 7.5.8, respectively. Because the ACL is security-critical information that cannot be reconstructed from information stored elsewhere, there is no utility to recover from a damaged `mailboxes` file. To protect the contents of the mailboxes, make frequent (even hourly) backups of the `mailboxes` file to some other part of the disk.
`user/`*`a...z`* directories	Contain user subscriptions. There is one file per user, each file containing a sorted list of the user's mailboxes. Each file name consists of the user's user name followed by a `.sub` file extension, and the files are sorted into the `a` through `z` directories by the first letter of the user name. There is no utility for recovering damaged subscription files. You can restore lost files from backups.
`proc` directory	Contains one file per active server process. The filename is the ASCII representation of the process id, and the file contains the following tab-separated fields: • Host name of client • Username, if logged in • Selected mailbox, if mailbox selected The `proc` subdirectory is normally purged when you reboot the server.

Table 7–2: Configuration Directory Contents (cont.)

File or Directory	Purpose
`quota/`*`a...z`* directories	Contain quota specifications for restricted IMAP users. There can be multiple files for each user, each file containing the limit for a quota root, as described in Section 7.5.9. Each file name consists of the string `user` followed by one or more extensions, starting with the associated user name (for example, user.hansen). The files are sorted into the `a` through `z` directories by the first letter of each user name. The `imapquota` program, when invoked with the `-f` switch, recalculates each user's quota. To remove the restrictions on a user's quota, remove the user's quota file. Then run `imapquota -f` to make the quota files consistent again.
`log` directory	Contains zero or more subdirectories, each named after a user. If a subdirectory exists for a user, the server keeps a telemetry log of protocol sessions authenticating as that user. The telemetry log is stored in the subdirectory with a filename that matches the server's process ID. Use this feature only for debugging purposes; the log files grow rapidly.

The largest database in the IMAP server is a user's mailbox directory. By default, these mailbox directories are located in the `/var/spool/imap/users` directory. There is one directory for each user and the directory name is the user's user name. If you have a highly populated mailbox tree, you can optionally sort these mailbox directories into `/var/spool/imap/users/`*`a...z`* subdirectories by specifying the `hashimapspool` option in the `imapd.conf` file. See `imapd.conf`(4) and `dohash`(8) for more information.

Each user's directory contains the files listed in Table 7–3.

Table 7–3: Mailbox Directory Contents

File or Directory	Purpose
message files	Contain one message each in RFC 822 format. Lines in the message are separated by a carriage return and line feed, not just a line feed. The file name of each message is the message's UID followed by a dot (.).
`cyrus.header`	Contains a magic number and variable-length information about the mailbox itself.
`cyrus.index`	Contains fixed-length information about the mailbox itself and each message in the mailbox.
`cyrus.cache`	Contains variable-length information about each message in the mailbox.
`cyrus.seen`	Contains variable-length state information about each user who has permission to read the mailbox.

The `reconstruct` utility can be used to recover from corruption in mailbox directories. If the `reconstruct` utility finds existing header and index files, it attempts to preserve any data in them that is not derivable from the message files themselves, including the flag names, flag state, and internal date. The utility derives all other information from the message files.

You can recover from a damaged disk by restoring message files from a backup and then running the `reconstruct` utility to regenerate what it can of the other files. The `reconstruct` program does not adjust the quota usage recorded in any quota file. After running `reconstruct`, run `imapquota -f` to fix the quota root files.

7.5.7 Mailbox Namespace

The IMAP server presents mailboxes using the `netnews` namespace convention. Mailbox names have the following restrictions:

- Are case-sensitive
- Cannot start or end with a period (.) character
- Cannot contain two period (..) characters in a row
- Cannot contain non-ASCII characters, shell metacharacters, or a backslash (/) character

All personal mailboxes for a user begin with the `user.`*`username`*`.` string. For example, mailboxes belonging to a user named Hansen begin with the `user.hansen.` string. If Hansen has a mailbox for work-related e-mail, it might be called `user.hansen.work`.

In the user's mail application, the prefix `user.hansen.` normally appears as `INBOX.`. The mailbox `user.hansen.work` therefore appears as `INBOX.work`. However, if the access control list (ACL) of the mailbox permits other users to see that mailbox, it appears to them as `user.hansen.work`.

You can create or delete a user's mailbox by creating or deleting the user's `INBOX`. A user with an `INBOX` can create and subscribe to personal mailboxes. Users with dots in their user names are able to log in, but cannot have an `INBOX` or receive IMAP mail. When you delete a user's `INBOX`, all of the personal mailboxes associated with it are deleted as well.

With the exception of `INBOX`, all mailbox names are system-wide; they refer to the same mailbox regardless of the user. ACLs determine which users can access or see certain mailboxes.

In contexts that permit relative mailbox names, the mailbox namespace works as follows:

- Names that do not start with a period (.) are fully qualified.
- Names that start with a period (.) are relative to the current context.

You might need to use this convention if you use the `telnet` command to connect to an IMAP port for troubleshooting purposes or if you create an application that issues IMAP calls.

If you are working with folder names and the top of the hierarchy is named `user.hansen`, the name `.work.personnel.issues` resolves to `user.hansen.work.personnel.issues` and the name `work.personnel.issues` resolves to `work.personnel.issues`.

7.5.8 Access Control Lists

Access to each mailbox is controlled by each mailbox's access control list (ACL). ACLs provide a mechanism for specifying the users or groups of users who have permission to access the mailboxes.

An ACL is a list of zero or more entries. Each entry has an identifier and a set of rights. The identifier specifies the user or group of users to which the entry applies. The set of rights is one or more letters or digits, each letter or digit conferring a particular privilege. See `cyradm`(1) for further information.

Access rights are defined as follows:

lookup (`l`)

The user can see that the mailbox exists.

read (r)

The user can read the mailbox. The user can select the mailbox, retrieve data, perform searches, and copy messages from the mailbox.

seen (s)

The per-user seen state is preserved. The server saves the `Seen` and `Recent` flags for the user.

write (w)

The user can modify flags and keywords other than `Seen` and `Deleted` (which are controlled by other sets of rights).

insert (i)

The user can insert new messages into the mailbox.

post (p)

The user can send mail to the submission address for the mailbox. This right differs from the `i` right in that the delivery system inserts trace information into submitted messages.

create (c)

The user can create new sub-mailboxes of the mailbox.

delete (d)

The user can store the `Deleted` flag, perform expunges, and delete.

administer (a)

The user can change the ACL on the mailbox.

You can combine access rights in different ways. For example:

lrs

The user can read the mailbox.

lrsp

The user can read the mailbox and can post to it through the delivery system. Most delivery systems do not provide authentication, so the post right usually has meaning only for the `anonymous` user.

lr

The user can see the mailbox and can read it, but the server does not preserve the `Seen` and `Recent` flags. This set of rights is useful primarily for `anonymous IMAP`.

rs

The user can read the mailbox and the server preserves the `Seen` and `Recent` flags, but the mailbox is not visible to the user through the various mailbox listing commands. The user must know the name of the mailbox to be able to access it.

lrsip

The user can read and append to the mailbox either through IMAP or through the delivery system.

Any identifier may be prefixed with a dash (-) character. The associated rights are then removed from that identifier. These are referred to as negative rights.

To calculate the set of rights granted to a user, the server first calculates the union of all rights granted to the user and to all groups of which the user is a member. The server then calculates and removes the union of all negative rights granted to the user and to all groups of which the user is a member. For example, in the following ACL, the user named Fred is granted the rights `lrswip` and the user anonymous is granted the rights `lrp`:

```
anyone        lrsp
fred          lwi
-anonymous    s
```

Regardless of the ACL on a mailbox, users who are listed in the `admins` configuration option of the `/etc/imapd.conf` file implicitly have the lookup and administer rights on all mailboxes. Users also implicitly have the lookup and administer rights on the `INBOX` and all of their personal mailboxes.

When a mailbox is created, its ACL starts with a copy of the ACL of its closest parent mailbox. When a user is created, the ACL on the user's `INBOX` starts with a single entry granting all rights to the user. When a nonuser mailbox is created and does not have a parent, its ACL is initialized to the value of the `defaultacl` option in the `/etc/imapd.conf` file.

7.5.9 Quotas

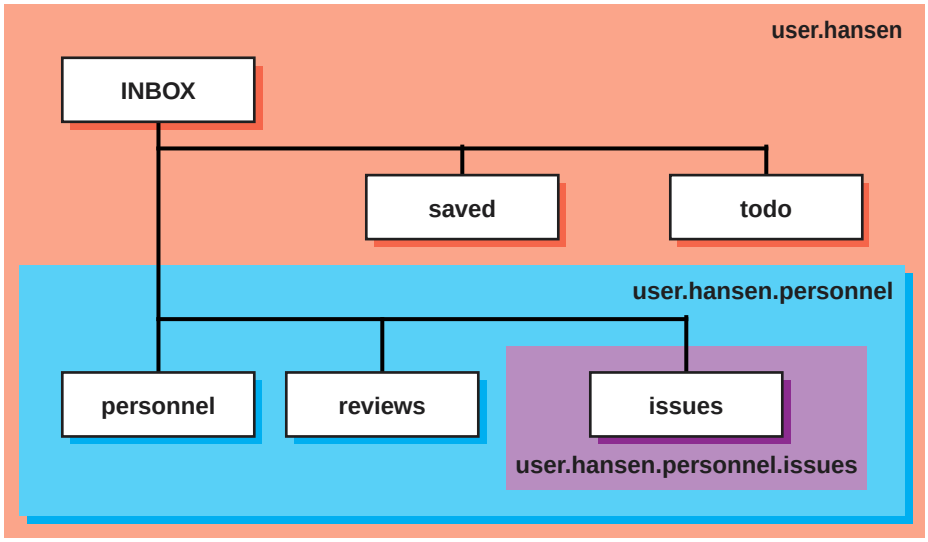
You can use quotas to limit the system resources available to a user. The IMAP server supports quotas on storage.

A quota on storage is defined as the number of kilobytes of disk space that a user's messages are permitted to consume. Each copy of a message is counted independently, even when the server can conserve disk space by making hard links to message files. The additional disk space overhead used by mailbox index and cache files is not charged against a quota.

You can assign one quota on the overall space permitted for a user's mailboxes or you can assign different quotas on selected branches of a user's mailbox hierarchy. In either case, you apply the quota to the root of the mailbox hierarchy that you want to limit. The quota root encompasses any number of mailboxes in that hierarchy. Quotas on a quota root apply to the sum of the usage by all mailboxes at that level and below that level that is not part of a quota root on lower level, hence, each mailbox is limited by at most one quota root.

Figure 7–7 shows an example of quota roots for a user named Hansen.

Figure 7–7: Quota Roots



ZK-1578U-AI

In Figure 7–7, the user Hansen has the following mail folders:

```
user.hansen (INBOX)
user.hansen.personnel
user.hansen.personnel.reviews
user.hansen.personnel.issues
user.hansen.saved
user.hansen.todo
```

The following quota roots in the quota/h directory restrict Hansen's disk usage:

```
user.hansen
user.hansen.personnel
user.hansen.personnel.issues
```

The quota root user.hansen applies to the INBOX, saved, and todo mail folders. The quota root user.hansen.personnel applies to the personnel and reviews mail folders. The quota root user.hansen.personnel.issues applies only to the issues mail folder. If the user.hansen.personnel and user.hansen.personnel.issues quota roots did not exist, the restrictions specified for the user.hansen root would apply to all mail folders in the user.hansen hierarchy (those mail folders with the user.hansen prefix).

You can create quota roots by using the setquota command in the cyradm utility; however, you cannot delete quota roots with this utility. To remove a quota root, you must remove the associated quota file.

For a message to be inserted into a mailbox, the mailbox must have sufficent storage so that inserting the message will not exceed the quota root. This is always true of manual transfers from one folder to another, but mail delivery is a special exception. If the limit is not exceeded when delivery starts, then the message is delivered regardless of its size. If delivery of the new message exceeds the folder's quota, the imapd daemon informs the user and permits him or her to correct the problem. If mail delivery were not permitted in this case, the user would not know that mail cannot be delivered.

When the quota root is exceeded, mail delivery fails with a temporary error. The system attempts delivery for a few days, providing the user time to notice and correct the problem.

When a user selects a mail folder that is near or exceeds the quota, the server issues an alert to notify the user. You can use the quotawarn configuration option to set the threshold of usage at which the server issues quota warnings. The server issues warnings only when the user has rights to the folder because only users with rights can correct the problem.

7.5.10 Partitions

You can use partitions to store mailboxes in different parts of your file system. Hierarchies of mailboxes can be spread across multiple disks. You must use the cyradm utility to specify these alternate partitions; you cannot specify them from an IMAP mail application.

When creating a new mailbox, specify the name of the partition for the mailbox as an argument to the createmailbox command in the cyradm

utility. If the partition is not specified, the mailbox inherits the partition of its parent mailbox. If the mailbox has no parent, it defaults to the partition specified in the `defaultpartition` configuration option.

You can also change the partition of an exisiting mailbox by using the `renamemailbox` command in the `cyradm` utility. See `cyradm`(8) for more information.

Note that quota roots are independent of partitions. A single quota root can apply to a mailbox hierarchy that spans multiple partitions.

7.6 Managing Mail

This section describes how to perform the following mail tasks:

- Monitor the mail queue
- Archive the mail queue
- Administer and Distribute Alias Information
- Display mail statistics

7.6.1 Monitoring the Mail Queue

Monitoring the mail queue enables you to determine the status of several types of networking operations, including jobs that have been queued on a local system for transfer to a remote system. General users and system administrators can monitor the mail queue.

To display the contents of the mail queue, use the `mailq` command. This command lists the number of requests and the queue ID, the message size, the date the message entered the queue, and the sender and recipient for each request. Alternatively, you can use the `sendmail –bp` command.

See `mailq`(1) for more information.

If a major host is off line for a period of time, the number of entries in the queue might be quite large, causing the performance of the mail environment to suffer. To remedy this, you must archive the queue. See Section 7.6.2 for information.

The following example shows two requests in the mail queue:

```
# mailq
               Mail Queue (2 requests)
--QID-- --Size-- -----Q-Time----- ------------Sender/Recipient------------
AA04956     1442 Tue Aug 24 10:12 <blaise>
                 (Deferred)
                                  <corcoran@host1.corp.com>
AA08618* (no control file)
```

7.6.2 Archiving the Mail Queue

When a major host is off line for a number of days, the mail queue might grow to be quite large. As a result, the `sendmail` utility spends a lot of time sorting the large queue, severely affecting the mail environment performance. Archiving the mail queue enables your mail environment to function normally while the major host is off line. To archive the mail queue, do the following:

1. Log in as root.
2. Change to the `/var/spool` directory by using the `cd` command.
3. Stop the `sendmail` utility by entering the following command:

   ```
   # /sbin/init.d/sendmail stop
   ```

4. Verify that the `sendmail` utility is not running by entering the following command:

   ```
   # ps -e | grep sendmail
   ```

5. Verify that no `sendmail` child processes are running by entering the following command:

   ```
   # ps -e | grep queue
   ```

   If any processes in the list are related to `sendmail`, for example, they include message queue IDs, it is best to wait until these processes are finished before moving the queue; otherwise, you might corrupt the queue data.
6. Move the `mqueue` directory to the `old.mqueue` directory by using the `mv` command.
7. Make a new `mqueue` directory by using the `mkdir` command.
8. Change the directory's permission code to 775 by using the `chmod` command.
9. Restart the `sendmail` utility by using the following command:

   ```
   # /sbin/init.d/sendmail restart
   ```

After the major host returns on line, process the old mail queue by using the following command:

```
# /usr/sbin/sendmail –oQ/var/spool/old.mqueue –q
```

When the queue is empty, remove it by using the following command:

```
# rm –r /var/spool/old.mqueue
```

7.6.3 Administering and Distributing Alias Information

Depending on how you choose to administer and distribute alias information on standalone or server systems, there are three ways to provide alias information for use in the mail environment:

- `/var/adm/sendmail/aliases` file
- NIS aliases database
- Lightweight Directory Access Protocol (LDAP)

By default, the `/var/adm/sendmail/aliases` file permissions code is 644. This means that global users cannot change and write the changes to the file. While this creates a reasonably secure system, it leaves the maintenance of the list of global users up to the system administrator.

You can distribute responsibility for maintenance by doing the following:

1. Create a local alias file for a global maintainer in a directory. Both the file and the directory must be accessible by another maintainer.
2. Create an entry in the `/var/adm/sendmail/aliases` file that includes the additional alias file. The entry has the following form:

   alias_name`:  :include:`*filename*

   The *filename* is the full path name and file name of the alias file.
3. Build a new version of the alias file by using the `newaliases` command.

See `aliases`(4) for more information.

Optionally, you can use NIS to administer and distribute alias information for use in the mail environment. To use the NIS aliases database, do the following:

1. Install and configure NIS, if this is not already done, by using the `nissetup` script.
2. Edit the `svc.conf` file by using the `svcsetup` script, and modify the aliases entry to include `yp` (NIS).
3. Edit the NIS aliases map to include the alias information you want.

See Chapter 3 for information on configuring NIS and Section 3.4.5 for information on updating an NIS map.

Lastly, you can also use LDAP to administer and distribute alias information for use in the mail environment. LDAP might be the best choice for maintaining alias information when your alias database is too large or you have many systems in the network sharing the same information.

To use LDAP for maintaining alias information, do the following:

1. Configure LDAP server and create the proper schema for your environment. You may configure LDAP service on the mail server or, preferably, on an independent system. See `sendmail.m4`(8) and your LDAP server documentation for more information.
2. Create two attributes in your schema: one for the user's mail address and another for the user's alias.
3. Manually edit the *`hostname`*`.m4` file in the `/var/adm/sendmail` directory and make the following changes:
   a. Set `_LDAPMap` to `{T}` to enable lookups.
   b. Set `_LDAPParam` to define the map and its argument list. See `sendmail.m4`(8) for details.
4. Switch to the `/var/adm/sendmail` directory and execute the following command:

   # **`make -f Makefile.cf.`*`hostname`***
5. Rename the *`hostname`*`.cf` file to `sendmail.cf`, as follows:

   # **`mv `*`hostname`*`.cf sendmail.cf`**
6. Restart the `sendmail` daemon by issuing the following command:

   # **`/sbin/init.d/sendmail restart`**

Note that you need to perform steps 3–6 each time you configure mail with the `mailsetup` or the `mailconfig` utilities.

7.6.4 Displaying Mail Statistics

You can display statistics about mail traffic on your system by using the `mailstats` command as follows:

`/usr/sbin/mailstats`

At any time, you can initialize the statistics file by issuing the following commands:

```
# cp /dev/null /var/adm/sendmail/sendmail.st
# chmod 666 /dev/null /var/adm/sendmail/sendmail.st
```

7.7 Mail Utilities

The operating system includes the following mail utilities:

- The `mail`, `binmail` utility (the default) — Used by the `sendmail` utility to deliver mail locally. Because the `mail` utility has root setuid permission, it handles delivery of all mail to a user's local mailbox

located in the /var/spool/mail directory. See the *Command and Shell User's Guide* and mail(1).

- The mailx, Mail utility — A combination of the Berkeley Software Distribution's (BSD) and UNIX System Laboratories, Inc.'s System V Release 4 (SVIDI) mail utilities. The mailx utility depends on the binmail utility for delivery to a user's mailbox. It has more user features than the binmail utility. See the *Command and Shell User's Guide* and mail(1).
- The dtmail utility — The default mail program in CDE. This utility uses sendmail as the transport and stores information in much the same way as the mailx utility. It also allows you to read POP3 mail, and offers support for MIME-encoded messages. See the *Common Desktop Environment: User's Guide* and dtmail(1).
- The message handler utility mh — It and its associated commands are included in the optional RAND Corporation Mail Handler subset (OSFMH). The message handler is composed of several shell commands where each command handles a specific function. For example, the inc command reads new mail and the comp command creates a message. Like the mailx utility, mh depends on the mail utility for delivery to a user's mailbox. The mh utility provides a graphical interface with the xmh command. See xmh(1X) for more information.
- Netscape Messenger — Part of the Netscape Communicator product, which is bundled with the operating system software. Messenger allows you to read mail from POP3 and IMAP4 mail servers. It also enables you to create rich HTML e-mail with embedded images, send MIME-encoded attachments, encrypt and decrypt your messages for privacy, use filters to organize your incoming messages into folders, and look up e-mail addresses. For more information on the Netscape Communicator product, see netscape(1).

For more information on sendmail, see sendmail(8), sendmail.cf(4), and sendmail.m4(8).

8

Simple Network Management Protocol

This chapter describes the Simple Network Management Protocol (SNMP) implementation on a Tru64 UNIX system.

The Simple Network Management Protocol (SNMP) is the de facto industry standard for managing Transmission Control Protocol/Internet Protocol (TCP/IP) networks. The protocol defines the role of a Network Management Station (NMS) and the SNMP Agent, allowing remote users on an NMS to monitor and manage TCP/IP network entities.

Note

Tru64 UNIX does not implement the NMS software.

Tru64 UNIX provides the `snmpd` daemon as the SNMP agent. This daemon is started at boot time. For information on how to set up and configure the `snmpd` daemon, see `snmpd`(8).

The operating system includes two SNMP subagents:

- `os_mibs` — Implements industry-standard management information base (MIB) support, including MIB II, the FDDI MIB, the Token Ring MIB, the Host Resources MIB, and an Ethernet-like Interfaces MIB. See `os_mibs`(8) for a list of the related RFCs and Appendix F for a description of the Host Resources MIB implementation.
- `cpq_mibs` — Implements MIBs that are specific to the Tru64 UNIX operating system. See `os_mibs`(8) for more information about these MIBs.

These subagents are started and stopped automatically in conjunction with the `snmpd` daemon. Together, they provide the SNMP data required by the Insight Manager daemon, `insightd`, for managing Tru64 UNIX systems via the web. For more information, see `insightd`(8) and `insight_manager`(5).

See the *Network Programmer's Guide* for information on registering applications with the SNMP agent.

9

Solving Network and Network Services Problems

This chapter contains a diagnostic map to help you solve problems that might occur when you use the network and network services software. Use this chapter together with the appropriate HP documentation to solve as many problems as possible at your level.

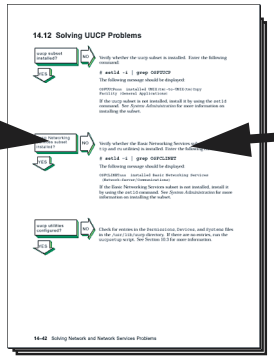
Section 9.1 and Section 9.2 provide information about how to use the diagnostic map and where in the map to start for certain problems. The sections that follow contain portions of the diagnostic map. They describe how to solve problems on the following systems:

- DNS/BIND Servers (Section 9.3)
- DNS/BIND Clients (Section 9.4)
- NIS Servers (Section 9.5)
- NIS Clients (Section 9.6)
- NFS Servers (Section 9.7)
- NFS Clients (Section 9.8)
- Systems running AutoFS (Section 9.9)
- Systems running UUCP (Section 9.10)
- Systems running NTP (Section 9.11)
- Systems running sendmail (Section 9.12)
- Systems running POP and IMAP mail (Section 9.13)

9.1 Using the Diagnostic Map

Network and network service problems can occur for a number of reasons. The diagnostic map in this chapter and a similar diagnostic map in the *Network Administration: Connections* manual help you to isolate the problem. The following figure explains how to use the diagnostic map:

The left-hand column asks questions about the status of specific events that occur as you use the system.



The right-hand column diagnoses negative responses to those questions.

After you isolate the problem, the map refers you to other chapters for instructions on using the various problem solving tools and utilities. The map also refers you to other manuals for more complete diagnostic information for particular devices and software products.

You could experience problems that are not documented in this manual when you use base system network and network services software with other layered products. See the documentation for the other products for additional information.

9.2 Getting Started

Before you start problem solving, ensure that the communications hardware is ready for use. Verify the following:

- The system's physical cable connections (the Ethernet connection and the transceiver connection) are properly installed. See the documentation for your system and communications hardware device.
- Event logging is enabled in order to monitor network events. See the *System Administration* manual for information on starting event logging and for descriptions of the event messages.

Also see the product release notes for up-to-date information on known problems.

Table 9–1 helps you identify a starting point in the diagnostic map.

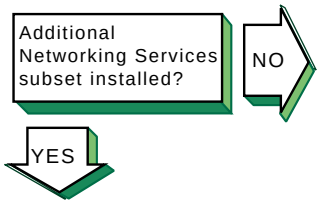
Table 9–1: Problem Solving Starting Points

If your problem is:	Start here:
uucp command error	Section 9.10

Table 9–1: Problem Solving Starting Points (cont.)

If your problem is:	Start here:
Network command error	Solving PPP Problems and Solving SLIP Problems sections in *Network Administration: Connections* Solving IPv4 Network Problems and Solving IPv6 Network Problems sections in *Network Administration: Connections*
Connecting to an ATM network	Solving ATM Problems section in *Network Administration: Connections* Solving IPv4 Network Problems and Solving IPv6 Network Problems sections in *Network Administration: Connections*
Obtaining an IP address using DHCP	Solving DHCP Problems section in *Network Administration: Connections* Solving IPv4 Network Problems and Solving IPv6 Network Problems sections in *Network Administration: Connections*
Correcting system time when you are using NTP	Section 9.11
Getting host name information	Section 9.4, if you are using DNS/BIND Section 9.6, if you are using NIS
Accessing files	Section 9.8, if you are using NFS Section 9.9, if you are using AutoFS Solving IPv4 Network Problems and Solving IPv6 Network Problems sections in *Network Administration: Connections*
Connecting to a host using LAT	Solving LAT Problems section in *Network Administration: Connections*.
Unknown errors	Solving IPv4 Network Problems section in *Network Administration: Connections*
Unknown IPv6 errors	Solving IPv6 Network Problems section in *Network Administration: Connections*
Sending or receiving mail	Section 9.12 Section 9.13, if you are using POP or IMAP mail

9.3 Solving DNS/BIND Server Problems



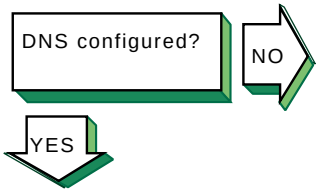
Verify that the Additional Networking Services subset is installed. Enter the following command:

```
# setld -i | grep OSFINET
```

If the subset is installed, the following message is displayed:

```
OSFINETnnn installed Additional Networking
Services  (Network-Server/Communications)
```

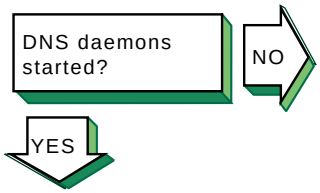
If the subset is not installed, install it by using the `setld` command. See the *Installation Guide* for more information on installing the subset.



Use the `rcmgr` utility to display the value of the `BIND_SERVERTYPE` entry in the `/etc/rc.config.common` file:

```
# rcmgr get BIND_SERVERTYPE
```

If no type is specified, run the SysMan Menu utility to configure your DNS server. See Section 2.5 for more information.

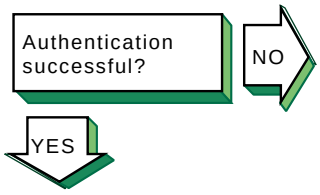


Verify that the BIND daemon (`named`) is running. Enter the following command:

```
# ps –e | grep named
```

If no `named` process is running, start the `named` daemon, using the following command:

```
# /sbin/init.d/named start
```

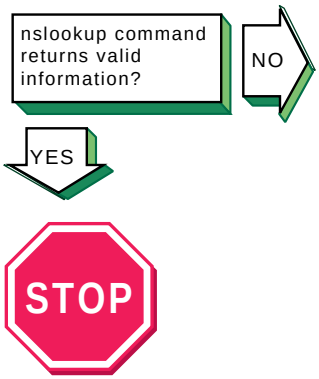


If you have enabled authentication, and secure dynamic updates or secure zone transfers are not successful, look for errors in the `daemon.log` file generated by the `syslogd` daemon. For secure dynamic updates, examine the log on the master server. For secure zone transfers, examine the log on the master server and the slave server. See Section 10.4 for more information about viewing `syslogd` message files.

If you see a `syntax error near 'item'` message, look for syntax errors in your `named.conf` file and key file (possibly `named.keys`). Verify that there are no missing braces, quotes, or semicolons. If necessary, compare the contents of these files with those in Section 2.6.3.

If you see an `unknown key` *`'key-name'`* message or an `Invalid TSIG secret "`*`key-string`*`"` message, do the following:

1. Verify that you are using the correct key for the update or transfer.
2. Verify the spelling of the key name.
3. Verify the integrity of the key string. There must be no line feeds or spaces between the quotes that contain the key.
4. Verify that the algorithm specified for the key is `hmac-md5` and that the key was generated correctly. If necessary, generate a new key. See Section 2.6 for more information.

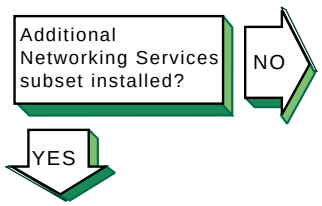


Problem still exists? Report it to your service representative. See Chapter 12.

If the `nslookup` command does not return information for any host or the host specified in the client `nslookup` command, use the value of the `BIND_SERVERTYPE` entry you collected in a previous step to select a course of action from below:

If the type is:	**Go to:**
`CLIENT`	Stop. This system is not a DNS/BIND server and cannot provide name resolution to clients.
`MASTER`	Section 11.4
`SLAVE`	Section 11.4
`FORWARDER`	Section 11.5
`CACHING`	Section 11.9

9.4 Solving DNS/BIND Client Problems



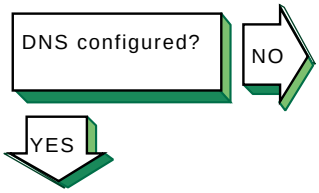
Verify that the Additional Networking Services subset is installed. Enter the following command:

```
# setld -i | grep OSFINET
```

If the subset is installed, the following message is displayed:

```
OSFINETnnn installed Additional Networking
Services  (Network-Server/Communications)
```

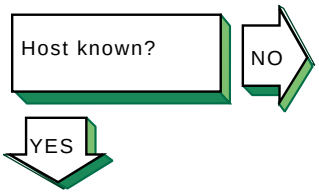
If the subset is not installed, install it by using the `setld` command. See the *Installation Guide* for more information on installing the subset.



Use the `rcmgr` utility to display the value of the `BIND_SERVERTYPE` entry in the `/etc/rc.config.common` file:

```
# rcmgr get BIND_SERVERTYPE
```

If no type is specified, run the SysMan Menu utility to configure your DNS client. See Section 2.5 for more information.



If you attempt to use one of the network commands (for example, `telnet`, `rlogin`, and `rsh` commands) and the remote host is not known, the following message is displayed:

```
unknown host
```

Complete the following steps:

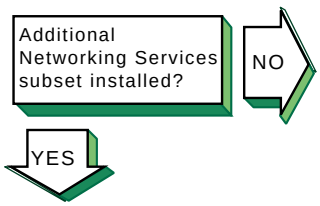
Problem still exists? Report it to your service representative. See Chapter 12.

1. Look in the `/etc/svc.conf` file to determine if DNS is being used for the `hosts` database lookup. If it is, go to step 2. If it is not, add it to the file by using the `/usr/sbin/svcsetup` script.
2. Retrieve information about the remote host with which you tried to communicate by using the `nslookup` command. Enter the following command:

   ```
   # nslookup hostname
   ```

   If the command succeeds, the client is set up correctly; try the network command again. If the command fails, go to step 3.
3. View the `/etc/resolv.conf` file and retrieve the addresses for the `nameserver` entries.
4. Verify that the servers are reachable by using the `ping` command. If no servers are reachable, contact your network administrator. If any name server fails to respond to the `ping` command, delete the name server entry from the `resolv.conf` file.
5. Try the `nslookup` command again. If the command fails, see the solutions for solving DNS/BIND server problems in Section 9.3.

9.5 Solving NIS Server Problems



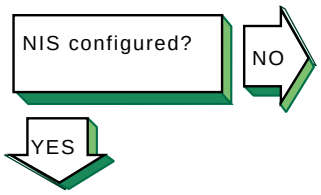
Verify that the Additional Networking Services subset is installed. Enter the following command:

```
# setld -i | grep OSFINET
```

If the subset is installed, the following message is displayed:

```
OSFINETnnn installed Additional Networking
Services  (Network-Server/Communications)
```

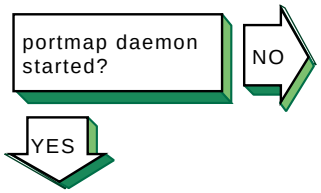
If the subset is not installed, install it by using the `setld` command. See the *Installation Guide* for more information on installing the subset.



Use the `rcmgr` utility to display the value of the `NIS_CONF` entry in the `/etc/rc.config.common` file:

```
# rcmgr get NIS_CONF
```

If nothing is returned, run the SysMan Menu utility to configure your NIS server. See Section 3.3 for more information.



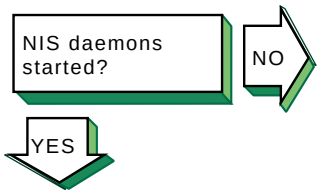
Verify that the `portmap` daemon is running. Enter the following command:

```
# ps -e | grep portmap
```

If you do not find the `portmap` daemon, stop and restart NIS, using the following commands:

```
# /sbin/init.d/nis stop
# /sbin/init.d/nis start
```

If the `portmap` daemon does not start, reboot the server.



Verify that a `ypserv` process is running. Enter the following command:

```
# ps -e | grep yp
```

If no `ypserv` process is running, stop and start NIS, using the following commands:

```
# /sbin/init.d/nis stop
# /sbin/init.d/nis start
```

If a `ypserv` process is running, execute a `ypwhich` command. Enter the following command:

```
# ypwhich
```

If nothing is returned, find the process ID (PID) of the `portmap` process and kill it. Enter the following commands:

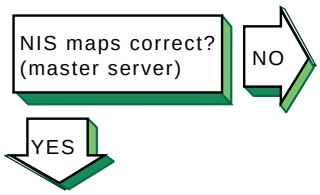
```
# ps -e | grep portmap
# kill -9 portmap_PID
```

Note

Because other network services use the `portmap` daemon, stopping it can affect network service. Therefore, notify your users of potential disruptions.

Stop and start NIS by using the following commands:

```
# /sbin/init.d/nis stop
# /sbin/init.d/nis start
```



Verify that the information in the map is correct. Enter the following command:

```
# ypcat map_name
```

The *map_name* variable is the name of the NIS map. If the information is incorrect, create a new map. Enter the following commands:

```
# cd /var/yp
# make map_name
```

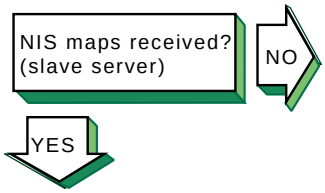
The following message is displayed:

map_name updated

If the make command indicates that the database is not updated, complete the following steps:

1. Remove the *database_name*.time file in the /var/yp and /var/yp/*domainname* directories.
2. Create a new map by using the make command. Enter the following commands:

   ```
   # cd /var/yp
   # make map_name
   ```



If you suspect that a slave server is not getting NIS map updates, complete the following steps on the slave server:

1. Verify that the NIS master server is running and reachable by using the ping command.
2. Create a ypxfr log file. Enter the following commands:

   ```
   # cd /var/yp
   # touch ypxfr.log
   ```

3. Run ypxfr interactively to get map updates. Enter the following command:

   ```
   # ypxfr mapname
   ```

4. Examine the ypxfr.log file and resolve any problems. Remove the log file to turn logging off.
5. Verify the ypxfr entries in the /var/spool/cron/crontabs/root file. Use either the pg command or the /usr/bin/crontab –l

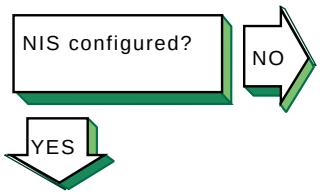
Problem still exists? Report it to your service representative. See Chapter 12.

command. The slave server entries are similar to the following:

```
# Network Information Service: SLAVE server entries
30 * * * * sh /var/yp/ypxfr_1perhour
31 1,13 * * * sh /var/yp/ypxfr_2perday
32 1 * * * sh /var/yp/ypxfr_2perday
```

6. Verify that the map has an entry in the corresponding `ypxfr` shell script.
7. Look in the `syslogd` daemon message files for any NIS messages. See Section 10.4 for more information.
8. Verify that the slave server is in the `ypservers` map for the domain.

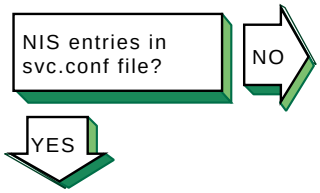
9.6 Solving NIS Client Problems



Use the `rcmgr` utility to display the value of the `NIS_CONF` entry in the `/etc/rc.config.common` file:

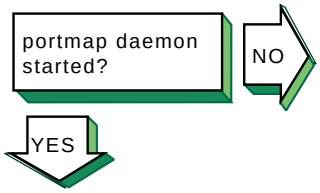
```
# rcmgr get NIS_CONF
```

If nothing is returned, run the SysMan Menu utility to configure your NIS client. See Section 3.3 for more information.



Use the `/usr/sbin/svcsetup` script to verify that the `svc.conf` file contains entries for NIS. NIS entries are indicated by the letters `yp`.

For the `passwd` and `group` databases, the Security Integration Architecture (SIA) controls whether or not NIS is used. However, in order to use NIS, a plus sign followed by a colon (`+:`) must be on the last line in both databases.



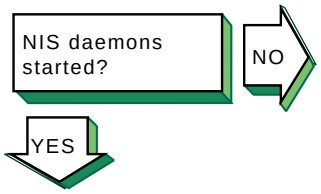
Verify that the `portmap` daemon is running. Enter the following command:

```
# ps -e | grep portmap
```

If no `portmap` daemon is running, stop and restart NIS, using the following commands:

```
# /sbin/init.d/nis stop
# /sbin/init.d/nis start
```

If the `portmap` daemon does not start, reboot the client.



Verify that a `ypbind` process is running. Enter the following command:

```
# ps -e | grep yp
```

If no `ypbind` process is running, stop and start NIS, using the following commands:

```
# /sbin/init.d/nis stop
# /sbin/init.d/nis start
```

If a `ypbind` process is running, enter the `ypwhich` command:

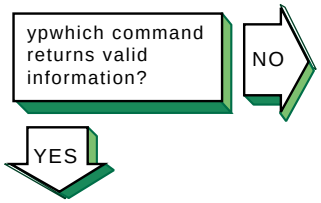
```
# ypwhich
```

If the `ypwhich` command does not return an answer, kill the `portmap` process. Enter the following command:

```
# kill -9 portmap_PID
```

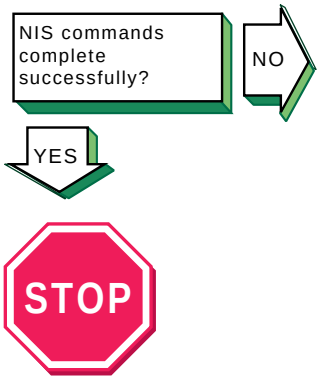
Stop and start NIS, using the following commands:

```
# /sbin/init.d/nis stop
# /sbin/init.d/nis start
```



If the `ypwhich` command displays inconsistent information when invoked several times in succession, your client system is changing the server system to which it is bound. This can occur over time, especially if your system is on a busy network or if the NIS servers are busy. Once all clients get acceptable response time from the NIS servers, the system will stabilize.

If the `ypwhich` command reports that the domain is not bound, your system did not initially bind to a server system. Issue a `ypcat` command, then reissue the `ypwhich` command.



Problem still exists? Report it to your service representative. See Chapter 12.

If an NIS command hangs, the following message is displayed on the console:

```
yp: server not responding for domain domainname.
Still trying
```

The client cannot communicate with the server. Complete the following steps:

1. Use the `rcmgr` command to verify that the domain name returned by the `domainname` command matches the value of the `NIS_DOMAIN` entry in the server's `/etc/rc.config.common` file:

   ```
   # rcmgr get NIS_CONF
   ```

   If the domain name does not match, or is not correct for your environment (note that the domain name is case-sensitive), reconfigure the client system by using the SysMan Menu utility. See Section 3.3 for more information.
2. Verify that at least one NIS server for your domain is running on your local subnetwork. If there is not, reconfigure the client by using the SysMan Menu utility, and choose to use the –S option to the `ypbind` command.
3. Determine if other clients on the subnetwork are having problems with any of the NIS commands.
4. Verify that `ypserv` daemon was started on the server by entering the following command:

   ```
   # rpcinfo -p server_name
   ```

   Also, verify that the `ypserv` daemon is currently running on the server by entering the following command:

   ```
   # rpcinfo -t server_name ypserv 2
   ```

   If the server fails either test, stop and restart NIS on the server as follows:

   ```
   # /sbin/init.d/nis stop
   # /sbin/init.d/nis start
   ```
5. Look in the `syslogd` daemon message files for any NIS messages. See Section 10.4 for more information.
6. Verify that the server is running. See the solutions for solving NIS server problems in Section 9.5.

If the previous steps do not solve the problem, complete the following steps:

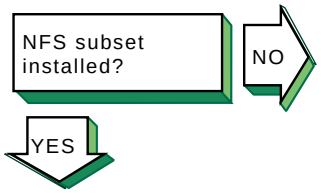
1. Stop and start NIS. Enter the following commands:

   ```
   # /sbin/init.d/nis stop
   # /sbin/init.d/nis start
   ```

If this does not solve the problem, go to step 2.

2. Reboot the system.
3. Reconfigure NIS by running the SysMan Menu utility.

9.7 Solving NFS Server Problems



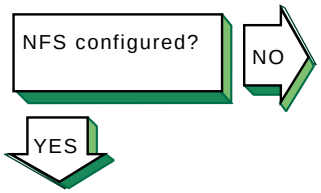
Verify that the NFS Utilities subset is installed. Enter the following command:

```
# setld –i | grep OSFNFS
```

If the subset is installed, the following message is displayed:

```
OSFNFSnnn  installed  NFS(tm) Utilities
(Network-Server/Communications)
```

If the subset is not installed, install it by using the `setld` command. See the *Installation Guide* for more information on installing the subset.

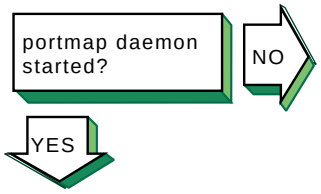


Use the `rcmgr` utility to display the value of the `NFSSERVING` entry in the `/etc/rc.config.common` file:

```
# rcmgr get NFSSERVING
```

If nothing is returned, run the SysMan Menu utility to configure your NFS server. See Section 4.3 for more information.

Verify that the network software has been configured. See the solution at Network configured? in the diagnostic map in *Network Administration: Connections*.



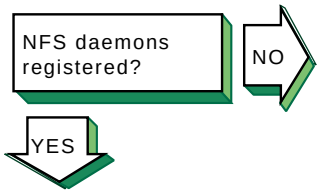
Verify that the `portmap` daemon is running. Enter the following command:

```
# ps –e | grep portmap
```

If the `portmap` daemon is not running, stop and restart NFS by using the following commands:

```
# /sbin/init.d/nfs stop
# /sbin/init.d/nfs start
```

If the `portmap` daemon does not start, reboot the server.

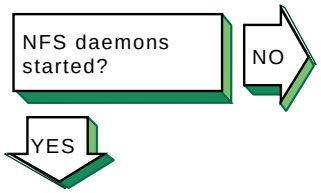


Verify that the NFS daemons are registered with the `portmap` daemon. Enter the following commands:

```
# rpcinfo –u server_name mount
# rpcinfo –u server_name nfs
```

If neither is registered, start NFS by using the following command:

```
# /sbin/init.d/nfs start
```



To verify that the NFS daemons are running, complete the following steps:

1. Verify that a `mountd` process is running. Enter the following command:

   ```
   # ps –e | grep mountd
   ```

   If a `mountd` process is running, go to step 2. If no `mountd` process is running, stop and start NFS by using the following commands:

   ```
   # /sbin/init.d/nfs stop
   # /sbin/init.d/nfs start
   ```

2. Verify that an `nfsd` process is running. Enter the following command:

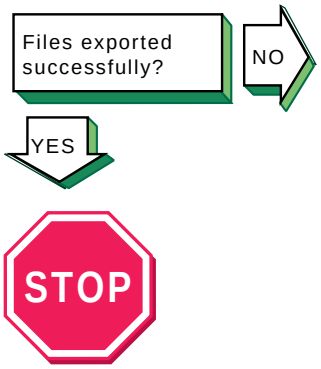
   ```
   # ps –e | grep nfsd
   ```

   If no `nfsd` process is running, stop and start NFS by using the following commands:

   ```
   # /sbin/init.d/nfs stop
   # /sbin/init.d/nfs start
   ```

Alternatively, you can use the SysMan Menu utility to view the status of some NFS daemons. You can skip directly to the status dialog box by entering the following command:

```
# /usr/sbin/sysman nfs_daemon_status
```



Problem still exists? Report it to your service representative. See Chapter 12.

To verify that the files are being exported, complete the following steps:

1. Verify that file is being exported. Enter the following command:

   ```
   # showmount –e
   ```

   If the file is being exported, go to step 3.

2. If the file is not being exported, verify that the file has an entry in the `/etc/exports` file. If there is no entry in the `/etc/exports` file, edit the file and create an entry. Have the remote system mount the file.

3. If the file is being exported and the users cannot mount the file, use the `rcmgr` utility to display the value of the `NONROOTMOUNTS` entry in the `/etc/rc.config` file and determine if the users are allowed to mount the file:

   ```
   # rcmgr get NONROOTMOUNTS
   ```

   If the `NONROOTMOUNTS` parameter is 0, only users running as root can mount files from this server. To allow users not running as root to mount the files, enter the following command:

   ```
   # rcmgr set NONROOTMOUNTS 1
   ```

4. Verify that the `mountd` daemon is running with Internet address checking on. Enter the following command:

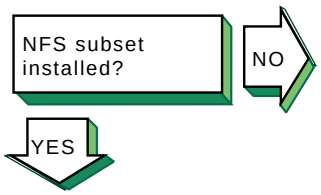
   ```
   # ps –e | grep mountd
   ```

   If the `–i` option is displayed, the client's name and address must be in the `/etc/hosts` file, or in the DNS or NIS `hosts` database. Only known hosts can mount the file system. If the `–d` or `–s` option is displayed, the client system must be in the same DNS domain or subdomain, respectively, as the server.

5. If the `mountd` daemon is returning stale file handles for exported files, send a hangup signal (SIGHUP) to the `mountd` daemon to force it to reread the `/etc/exports` file. Enter the following commands:

   ```
   # ps –e | grep mountd
   # kill -1 mountd_pid
   ```

9.8 Solving NFS Client Problems



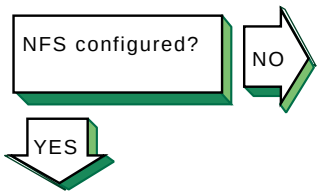
Verify that the NFS Utilities subset is installed. Enter the following command:

```
# setld –i | grep OSFNFS
```

If the subset is installed, the following message is displayed:

```
OSFNFSnnn  installed  NFS(tm) Utilities
(Network-Server/Communications)
```

If the subset is not installed, install it by using the `setld` command. See the *Installation Guide* for more information on installing the subset.

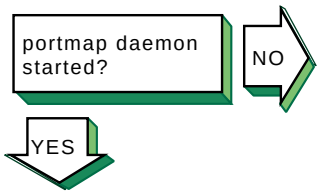


Use the `rcmgr` utility to display the value of the `NFS_CONFIGURED` entry in the `/etc/rc.config.common` file:

```
# rcmgr get NFS_CONFIGURED
```

If nothing is returned, run the SysMan Menu utility to configure your NFS client. See Section 4.3 for more information.

Verify that the network software has been configured. See the solution at Network configured? in the diagnostic map in *Network Administration: Connections*.



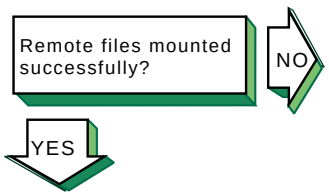
Verify that the `portmap` daemon is running. Enter the following command:

```
# ps –e | grep portmap
```

If the `portmap` daemon is not running, stop and restart NFS by using the following commands:

```
# /sbin/init.d/nfs stop
# /sbin/init.d/nfs start
```

If the `portmap` daemon does not start, reboot the client.



If the client cannot mount a remote file system or directory, complete the following steps:

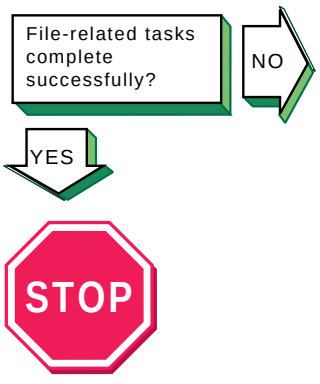
1. If an error message is displayed on the user's terminal, see Appendix C for the error message and a description.
2. Verify that the remote NFS server is on your local network and in your `hosts` database.
3. Verify that the server daemons on the remote system are running. Enter the following command:

   ```
   # rpcinfo –p server_name
   ```

4. Verify that the server is exporting the files you want to mount. Enter the following command:

   # **showmount –e *server_name***

5. See the solutions for solving NFS server problems in Section 9.7. If the server is running and you still have problems, verify the Ethernet connections and the Internet connections between the client system and the remote server.
6. Determine if other clients on the network are having problems with the remote server.
7. Verify that the mount command line or the entry in the `/etc/fstab` file is correct, and verify the following:
   a. The host name matches the name of the remote NFS server.
   b. The mount point exists on your system.
8. If you get an authentication error, verify the following:
   a. If you are not a superuser, the server allows nonroot mounts.
   b. Your host name is in the server's `hosts` database.
   c. If your system is not in the same domain as the server, the server performs domain checking. See `mountd`(8) for more information on server options.



Problem still exists? Report it to your service representative. See Chapter 12.

If application programs that perform file-related tasks do not complete their tasks or take a long time to do so, complete the following steps:

1. If an error message is displayed on the user's terminal, see Appendix C for the error message and a description.
2. Verify that the server is running. See the solutions for solving NFS server problems in Section 9.7. If the server is running, verify that the `nfsd` daemon is accumulating CPU time. If it is not, kill it and restart it. If this does not solve the problem, reboot the server. If the remote file systems or directories are mounted with the `hard` option, the program continues when the server is running once again.
3. Determine if other clients on the network are having problems with the remote server. If they are not, verify that the Ethernet connections and the internet connections between the client system and the remote server are working properly.
4. Determine if any `nfsiod` daemons are running. Enter the following command:

```
# ps –e | grep nfsiod
```

If no `nfsiod` daemons are running, start some. Enter the following command:

```
# /usr/sbin/nfsiod 7
```

Although the `nfsiod` daemons are not necessary for a client, they perform read-ahead and write-behind functions, which might make I/O faster.

5. If file access requests succeed but file locking requests hang indefinitely, verify that the local `rpc.statd` and `rpc.lockd` daemons are running. Enter the following commands:

```
# ps –e | grep rpc.statd
# ps –e | grep rpc.lockd
```

If they are not running, start them. Enter the following commands:

```
# /usr/sbin/rpc.statd
# /usr/sbin/rpc.lockd
```

Also, verify that the local `rpc.statd` and `rpc.lockd` daemons are running on the server. Enter the following commands:

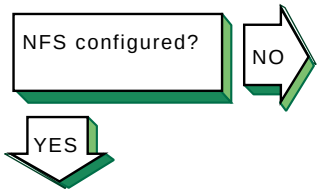
```
# rpcinfo –p server_name | grep status
# rpcinfo –p server_name | grep lockmgr
```

If they are not running, contact the server's system administrator.

Alternatively, you can use the SysMan Menu utility to view the status of some NFS daemons. You can skip directly to the status dialog box by entering the following command:

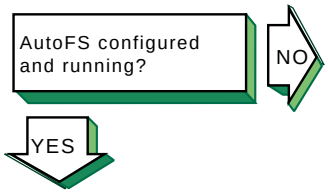
```
# /usr/sbin/sysman nfs_daemon_status
```

9.9 Solving AutoFS Problems



Verify that NFS is configured and functioning properly on the server and the client systems. AutoFS requires NFS to serve file systems.

Section 9.7 and Section 9.8 for basic NFS troubleshooting information.



To verify that the AutoFS service is configured and running, do the following:

1. Use the `rcmgr` utility to display the values of the `AUTOFS`, `AUTOFSD_ARGS`, and `AUTOFSMOUNT_ARGS` parameters in the `/etc/rc.config.common` file:

   ```
   # rcmgr -c get parameter
   ```

   Verify that the parameters are set as follows:

   - The `AUTOFS` parameter must be set to 1, indicating that it is to be started whenever you reboot the system or restart the NFS service.
   - The `AUTOFSD_ARGS` and `AUTOFSMOUNT_ARGS` parameters must be configured with the appropriate arguments for the `autofsd` daemon and `autofsmount` command. See Section 4.6.3.2, Section 4.6.3.4, `autofsd`(8), and `autofsmount`(8) for more information.

   Note

   If you use the `-D` option to define environment variables for AutoFS, you must define the same variables in the `AUTOFSD_ARGS` and `AUTOFSMOUNT_ARGS` parameters.

   If necessary, use the `rcmgr` utility to change any incorrect parameters, as follows:

   ```
   # rcmgr -c set parameter "value"
   ```

   Then, stop and restart NFS by using the following commands:

   ```
   # /sbin/init.d/nfs stop
   # /sbin/init.d/nfs start
   ```

   You can subsequently verify that the `/usr/bin/autofsd` daemon is running by entering the following command:

   ```
   # ps –e | grep autofsd
   ```

2. Verify that the `/usr/sbin/automount` daemon is not running by entering the following command:

   ```
   # ps –e | grep automount
   ```

   You cannot run AutoFS and Automount together on the same system. If the `automount` daemon is running, terminate the associated process, as follows:

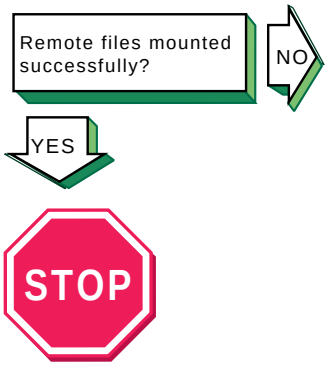
   ```
   # kill -TERM automount-pid
   ```

Also, update the value of the AUTOMOUNT parameter in the rc.config.common file to indicate that Automount is not to be started:

```
# rcmgr -c set AUTOMOUNT 0
```

Then, stop and restart NFS by using the following commands:

```
# /sbin/init.d/nfs stop
# /sbin/init.d/nfs start
```



Problem still exists? Report it to your service representative. See Chapter 12.

If an expected AutoFS auto-mount does not occur, complete the following steps:

1. Examine the contents of the daemon.log and user.log files for error messages related to AutoFS. See Section 10.4 for information on viewing the syslogd message files.

   The most common errors reported by the autofsd daemon concern insufficient export permissions and timeouts due to unresponsive servers. Again, see Section 9.7 and Section 9.8 for basic NFS troubleshooting information.

2. Verify that the AutoFS intercept point for the auto-mount in question exists.

   Enter the following command to list all of the AutoFS intercept points on your system:

   ```
   # mount -e | grep autofs
   ```

   If an expected intercept point is not defined:

   Note

   Only one intercept point is associated with any indirect map.

   a. If the associated AutoFS map is provided by NIS, enter the following command to verify that NIS has the latest version of the map:

      ```
      # ypcat -d domain mapname
      ```

      If the ypcat command displays old map file entries, update and redistribute the map by following the instructions in Section 3.4.5.

      If there is no response to the ypcat command, see Section 9.5 and Section 9.6 for information about troubleshooting NIS.

   b. Verify the syntax and spelling of the associated entries in your local or NIS-distributed map file. See Appendix A for more information about map file syntax.

c. Verify that you have removed the corresponding mount or symbolic link for any AutoFS map entry that you recently modified or removed. AutoFS cannot auto-mount a file system on a intercept point that is occupied by an active mount or symbolic link.

   Enter the following command to list the NFS file systems currently mounted on your system:

   ```
   # mount -e -t nfs
   ```

   Remove any AutoFS mount on this list that interferes with the intercept point. Use the following command:

   ```
   # autofsmount -t directory
   ```

   Symbolic links are not displayed by the `mount` command; therefore, you need to manually search for the symbolic links that AutoFS generates. If necessary, you can examine your AutoFS map entries to determine which file systems are served by a symbolic link, then use the `ls -l` command to find these links. (See Section A.4 for more information about AutoFS behavior.)

   If you find a symbolic link that is associated with an auto-mount, you can remove it by executing the `rm` command, as follows:

   ```
   # rm link
   ```

d. Verify that you removed any NFS file systems you might have manually mounted on the intercept point.

   Enter the following command to list the NFS file systems currently mounted on your system:

   ```
   # mount -e -t nfs
   ```

   Remove any manual mount on this list that interferes with the intercept point. Use the `umount` command and edit the `/etc/fstab` file, as necessary.

When you are finished verifying your map files and clearing AutoFS intercept points, execute the `autofsmount` command with the appropriate arguments to apply the changes. If you defined the arguments in the `AUTOFSMOUNT_ARGS` parameter of the `rc.config.common` file, you can execute the following command:

```
# /usr/sbin/autofsmount 'rcmgr -c get
AUTOFSMOUNT_ARGS'
```

Then, execute the `mount -e` command to verify that the intended intercept point exists.

3. If the problem persists, remove all AutoFS-related intercepts and auto-mounted file systems for the associated map file by entering the following command:

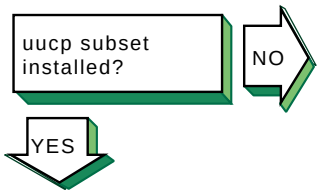
   # **/usr/sbin/autofsmount -M *mapname***

   Note

   Busy file systems are not unmounted; users must release them first.

   Then, execute the `autofsmount` and `mount -e` commands as before to process the maps and verify that the intended intercept point exists.

9.10 Solving UUCP Problems



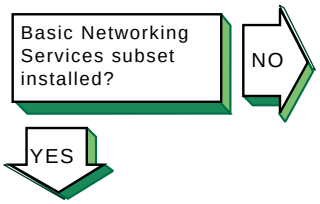
Verify that the UNIX-to-UNIX Copy Facility subset is installed. Enter the following command:

setld -i | grep OSFUUCP

If the subset is installed, the following message is displayed:

```
OSFUUCPnnn  installed UNIX(tm)-to-UNIX(tm) Copy
Facility (General Applications)
```

If the subset is not installed, install it by using the `setld` command. See the *Installation Guide* for more information on installing the subset.



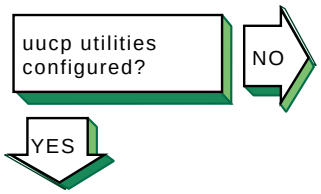
Verify that the Basic Networking Services subset (containing the `tip` and `cu` utilities) is installed. Enter the following command:

setld –i | grep OSFCLINET

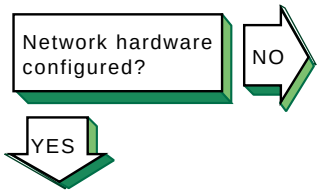
If the subset is installed, the following message is displayed:

```
OSFCLINETnnn  installed Basic Networking Services
(Network-Server/Communications)
```

If the subset is not installed, install it by using the `setld` command. See the *Installation Guide* for more information on installing the subset.

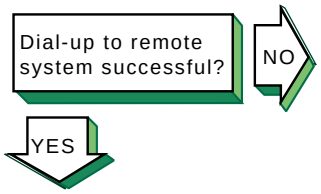


Look for entries in the `Permissions`, `Devices`, and `Systems` files in the `/usr/lib/uucp` directory. If there are no entries, run the `uucpsetup` script. See Section 5.3 for more information.



Configure the network hardware as follows:

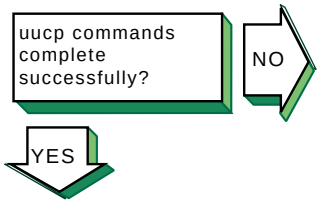
- Direct connections to remote host — Use a null modem or modem eliminator cable to connect your system to the remote host.
- Phone line connection to remote host — Use a cable to connect your system to a modem and another cable to connect your modem to a phone line. The modem you use must be compatible with the modem at the remote host. Make sure the modem is configured as follows:
  - Forced data set ready (DSR) is disabled.
  - Full or verbose status messages are enabled.
  - Character echo is disabled.
  - Use 8-bit characters with no parity.
  - XON/XOFF flow control is disabled.
- TCP/IP connection to remote host — Use a cable to connect your system to the network. Then, run the Network Configuration application to configure the network. See *Network Administration: Connections* for more information on setting up the network.



If you cannot dial up the remote system, verify the following:

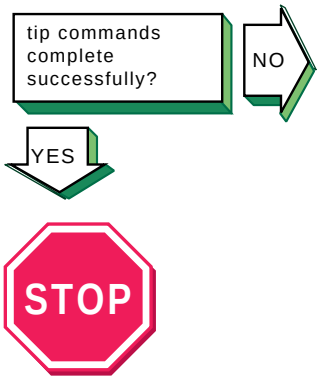
1. Make sure that the setup parameters (such as speed, parity, modem control, flow control, and other terminal characteristics) on the local and remote ends are properly defined for your modem type.
2. Dial the number to the remote node. If you do not get an `Attached` message or a login prompt, plug a telephone handset into the local telephone line to verify that there is a dial tone. If you do not hear a dial tone, call you local carrier to fix this problem. If you get no message, verify that the cabling between the local system and the modem is properly installed and undamaged.
3. If you get a dial tone, check that your modem is operational and perform diagnostic tests on your modem. See the modem manual for more information.
4. From another handset, dial the local telephone line. If the local telephone rings and you can carry on a conversation, the telephone line on the local end is good. If you cannot pass voice traffic, or if there is no ring, call your local carrier to fix this problem.
5. Repeat steps 2 and 3 on the remote node to resolve problems with the remote end.

6. If the telephone line is operational, verify that the remote modem is set up to automatically answer incoming calls when the system raises the data terminal ready (DTR) signal. The system raises the DTR signal by issuing a `uugetty` or `getty` command on the port.



Run the `uucp` tests to test the connection to the remote system. See Section 10.1 and Section 10.2.

If you can establish a connection, but your file transfer eventually times out and exits, attempt to set the type of flow control that the `uucico` daemon uses, as described in Section 5.3.5 and the `uucico`(8) reference page.

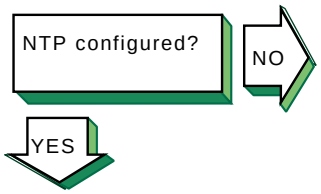


If the `tip` command does not execute successfully, complete the following steps:

1. Verify that the system name, connection speed, and phone number are in the `/etc/remote` file or that the system name and connection speed are in the `/etc/remote` file and the phone number is in the `/etc/phones` file. See `remote`(4) and `phones`(4) for more information.
2. Examine the `at` entry in the `/etc/remote` file. If the entry is correct, create an entry for the modem in the `/etc/acucap` file. See `acucap`(4) for more information.
3. Verify that the remote system is configured to answer incoming calls.

Problem still exists? Report it to your service representative. See Chapter 12.

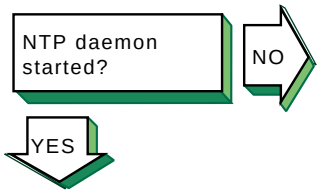
9.11 Solving NTP Problems



Use the `rcmgr` utility to display the value of the `XNTPD_CONF` entry in the `/etc/rc.config` file:

```
# rcmgr get XNTPD_CONF
```

If nothing is returned, run the SysMan Menu utility to configure NTP. See Section 6.3 for more information.



Verify that an `xntpd` process is running. Enter the following command:

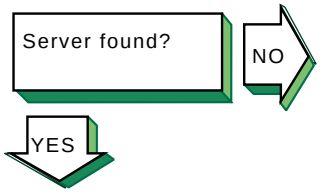
```
# ps –e | grep xntpd
```

Alternatively, you can use the SysMan Menu utility to view the status of the `xntpd` daemon. You can skip directly to the status dialog box by entering the following command:

```
# /usr/sbin/sysman ntp_status
```

If no `xntpd` process is running, start NTP by using the following command:

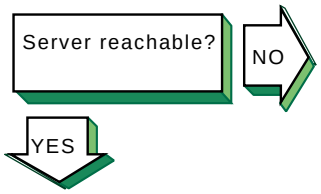
```
# /sbin/init.d/xntpd start
```



If the `ntpq` or `xntpdc` command cannot find the server host, the following message is displayed:

```
***Can't find host hostname
```

The *hostname* is not in the `/etc/hosts` file, the DNS `hosts` database, or the NIS `hosts` database. Edit the appropriate file or database and add an entry for the server host.



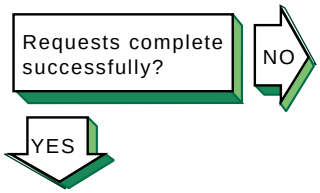
If you run one of the monitor programs and in the output from the `peers` command the reach column contains zeros (0s), complete the following steps:

1. Contact the system administrator of the server and verify which NTP daemon the server is running. The entry for the server in the `/etc/ntp.conf` file must contain the phrase `version` *x* after the server name, as follows:

   ```
   server host1 version x
   ```

2. Look in the `/etc/hosts` file and verify that there is an entry for each NTP server specified in the `/etc/ntp.conf` file. If you are using either DNS or NIS for host information, verify that the `hosts` database has an entry for each NTP server.

If the `xntpdc` *hostname* command does not display any information, verify that the *hostname* server is running NTP.



Problem still exists? Report it to your service representative. See Chapter 12.

If the `ntpq` or `xntpdc` request times out, the following message is displayed:

```
hostname: timed out, nothing received
***Request timed out
```

Complete the following steps:

1. The *hostname* is not running the `xntpd` daemon. Contact the system administrator for that system.
2. The network connection has gone down. See the solutions for Host Reachable? in the diagnostic map in *Network Administration: Connections*.

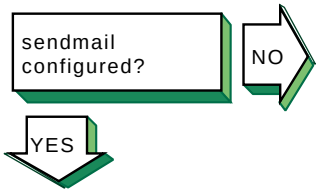
If you still cannot solve the problem, complete the following steps:

1. Examine the `/etc/rc.config` file to make sure it contains entries similar to the following:

   ```
   XNTPD_CONF="YES"
   export XNTPD_CONF
   XNTP_SERV1=server1
   export XNTP_SERV1
   XNTP_SERV2=server2
   export XNTP_SERV2
   XNTP_SERV3=server3
   export XNTP_SERV3
   XNTPD_OPTS="-g"
   export XNTPD_OPTS
   ```

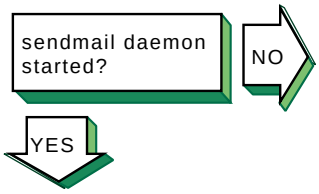
   If this entry does not exist or is incorrect, run the SysMan Menu utility to configure NTP. See Section 6.3 for more information.
2. Examine the `/etc/ntp.conf` file and make sure the information in it is accurate. It must contain entries for hosts running NTP with which you want to synchronize system time. Make sure the correct version number is specified for each server and peer. Use the SysMan Menu utility to correct any entries. See Section 6.3 for information.
3. Look in the `/var/adm/syslog.dated/current/daemon.log` file for information about NTP problems on the system. See Section 10.4 for more information.

9.12 Solving sendmail Problems



Verify that mail is configured by switching to the `/var/adm/sendmail` directory and checking for the presence of the `sendmail.cf` and `sendmail.cf.orig` files.

If one of the files does not exist, run the SysMan Menu utility to configure mail. See Section 6.3 for more information.

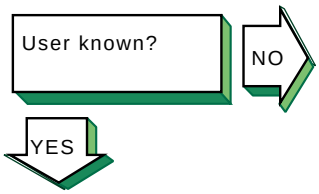


Verify that the `sendmail` command has been started. Enter the following command:

```
# ps –e | grep sendmail
```

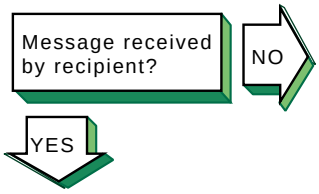
If `sendmail` is not running, start it using the following command:

```
# /sbin/init.d/sendmail start
```



If a user cannot send mail to another user, complete the following steps:

1. Determine if the `aliases` database was changed. If it was, update the database by using the `newaliases` command.
2. Look in the `mail.log` files generated by the `syslogd` daemon for the specific mail message. If the message reached its destination, the addressee is not on the destination system. Verify that the user has the correct address. See Section 10.4 for information on viewing the `syslogd` message files.



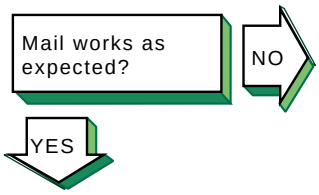
If you sent a mail message and the recipient did not receive it, complete the following steps:

1. Verify that the address is correct.
2. Verify that the remote node is reachable by using the `ping` command.
3. Look in the `mail.log` files generated by the `syslogd` daemon for the sender's user name. See Section 10.4 for information on viewing the `syslogd` message files. If you find an entry, write down the message ID. If no entry is found, send the message again.
4. Using the message ID, search through the `mail.log` files for the "from" and "to" entries. If you find a "from" entry but no "to" entry, either `sendmail` did not receive the message or the message was corrupted. Look in the `/var/spool/mqueue` directory for files containing the message ID by entering the following command:

```
# ls –l /var/spool/mqueue/*fmessage_ID
```

Possible outcomes include:

- The `qf*`*`message_ID`* control file is present but the (`df*`*`message_ID`*) data file is not. The message was lost.
- A "from" entry and a "to" entry exist, and the status is deferred. The message is in the queue.
- There is no corresponding sent entry. Use the `mailq` command to report the reason for the deferral.
- A "from" entry and a "to" entry exist, the status is sent, and the message was delivered. If a local delivery, the message reached the destination. If a remote delivery, have the system administrator on the remote host search for the message.



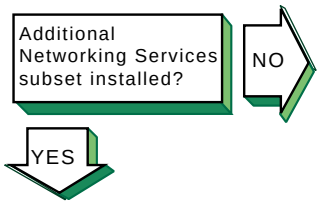
If sendmail is not working correctly, complete the following steps:

1. Look in the rejected message for an error message.
2. Look for error messages in the `mail.log` files generated by the `syslogd` daemon. See Section 10.4 for information on viewing the `syslogd` message files.

See Appendix E for a list of `sendmail` error messages.

Problem still exists? Report it to your service representative. See Chapter 12.

9.13 Solving POP and IMAP Problems



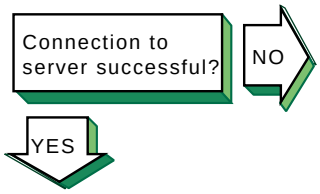
Verify that the Additional Networking Services subset is installed on the server. Enter the following command:

```
# setld -i | grep OSFINET
```

If the subset is installed, the following message is displayed:

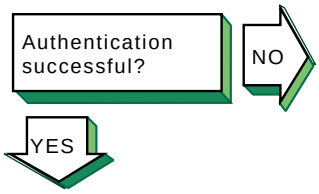
```
OSFINETnnn installed Additional Networking
Services  (Network-Server/Communications)
```

If the subset is not installed, install it by using the `setld` command. See the *Installation Guide* for more information on installing the subset.

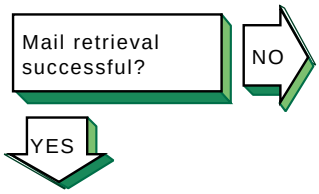


If the user cannot connect to the Post Office Protocol (POP) or Internet Message Access Protocol (IMAP) server:

1. Verify that the user is connecting to the correct server.
2. Verify that the server is reachable by using the `ping` command.
3. Verify the POP or IMAP entries in the `/etc/passwd`, `/etc/services`, and `/etc/inetd.conf` files on the server, as described in Section 7.4.1 and Section 7.5.1. If necessary, restart network services to effect the changes.

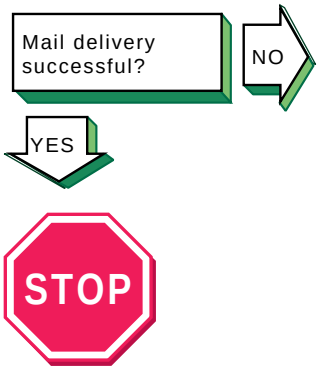


Verify that the user has specified a valid user name and password. Use the `mailusradm` utility to verify the existence of the POP or IMAP account on the server or to change the password, if necessary.



If a user cannot retrieve mail from the POP or IMAP server:

1. Verify that the user has a POP3 or IMAP4-compatible mail program.
2. For POP, look in the `/usr/spool/mail` directory for a lock file named after the user. If one exists, delete the file to remove the lock.
3. For IMAP, verify that the user has proper ACLs for the IMAP mail folder by using the `cyradm` command. See Section 7.5.8 and `cyradm`(8).
4. Look in the `mail.log` files generated by the `syslogd` daemon for error messages related to POP or IMAP. See Section 10.4 for information on viewing the `syslogd` message files.
5. Create a directory with the user's account name in the *`configdirectory`*`/log` directory (usually, the log directory is `/var/imap/log`, see the `/etc/imapd.conf` file for the location of the *`configdirectory`* on your system). When the user attempts to access the server, examine the log of the session to see where the error occurs.



Problem still exists? Report it to your service representative. See Chapter 12.

If the user is not receiving new mail:

1. Look in the `mail.log` files generated by the `syslogd` daemon for errors. See Section 10.4 for information on viewing the `syslogd` message files.
2. For IMAP, look at the `user` and `quota` configuration directories to verify that subdirectories `a` through `z` exist (see Section 7.5.2), that the subdirectories contain the proper files for the given user (see Section 7.5.6), and that all directories and files under `/var/imap` and `/var/spool/imap` are owned by the `imap` user.

If the user cannot send mail:

1. Verify that the user is connecting to the correct SMTP server.
2. Verify that the SMTP server is reachable by using the `ping` command.
3. See Section 9.12 on solving `sendmail` problems.

10

Using the Problem Solving Tools

To help you resolve problems with network services, the operating system provides problem solving tools you can use to complete the following tasks:

- Test a UUCP remote connection (Section 10.1)
- Monitor a UUCP file transfer (Section 10.2)
- Display the error log file (Section 10.3)
- Display the `syslogd` daemon message files (Section 10.4)

The following sections contain information about using the tools associated with these tasks. For information about additional tools you can use to diagnose network connections and network hardware, see *Network Administration: Connections*.

10.1 Testing a UUCP Remote Connection

Testing a UUCP remote connection can help you diagnose certain UUCP problems; for example, to determine why there is a backlog of transfer requests in the queue.

To test a remote connection, do the following:

1. Log in as root.
2. Change to the `/usr/lib/uucp` directory by using the `cd` command.
3. Test the remote connection by using the `uutry` command, using the following syntax:

   uutry *system_name*

   The *system_name* variable names the remote system to contact.
4. Examine the debugging output; the last line contains the status of the transaction. If your local system establishes a connection to the remote system, the debugging output contains a good deal of information. You can press Ctrl/c to stop the `uutry` shell script.

The `uutry` command has the following characteristics:

- It is a shell script stored in the `/usr/lib/uucp` directory.
- It contacts a remote system with debugging turned on. If you are using the UUCP scheduler, `uusched`, to start the `uucico` daemon automatically at

specified intervals, the uutry command overrides the retry time interval specified in the /usr/spool/uucp/.Status/*system_name* file.

If you use the uutry command frequently, you can put the pathname to the command in the PATH entry in your .profile file.

- It directs debugging information to a file named /tmp/*system_name*, where *system_name* is the name of the local system. The uutry command then executes a tail –f command to display the file's contents.

If your local system cannot contact the remote system, do the following:

1. Validate the physical connections between the local and remote systems. At both systems, confirm that the computer is turned on, that all the cables are properly connected, that the ports are enabled, and the modems (if being used) are working. If the remote system is not at your physical location, contact the administrator of the remote system.
2. Verify all configuration files on both systems. Verify that all entries in the Devices, Systems, and Permissions files are correct. If you are using a modem, verify all entries in the Dialers and Dialcodes files.

   If you are using a TCP/IP connection, verify that the configuration files contain the correct TCP entries. Verify that the inetd daemon can start the uucpd daemon. Edit the /etc/inetd.conf file and delete the comment character (#) from the beginning of the line containing the uucp entry. Restart the inetd daemon by using the following command:

   ```
   # /sbin/init.d/inetd start
   ```

Always save the debugging output produced by the uutry command until you are certain that the problem is resolved.

The following example shows a successful test of a remote connection to system host6:

```
# /usr/lib/uucp/uutry host6
   ⋮
Conversation Complete: Status SUCCEEDED
```

The following example shows an unsuccessful test of a remote connection to system host6:

```
# /usr/lib/uucp/uutry host6
   ⋮
mchFind called (host6)
conn (host6)
getto ret -1
Call Failed: CAN'T ACCESS DEVICE
exit code 101
```

```
Conversation Complete: Status FAILED
```

10.2 Monitoring a UUCP File Transfer

Monitoring a UUCP file transfer enables you to diagnose other UUCP problems, especially if you can already establish a remote UUCP connection.

To monitor a file transfer, do the following:

1. Verify the status of the files in the spooling directory on your local system by using the `uustat –q` command.
2. Verify that the local system can contact the remote system by using the `uutry` *system_name* command.
3. If the debugging output indicates that the connection was not successful, follow the steps described in Section 10.1 to test the remote connection..
4. Prepare a file for transfer by using the `uucp –r` command. The `–r` option instructs the `uucp` utility to place the file in the queue without starting the `uucico` daemon.

   Start the file transfer by using the `uutry` command.

See `uutry`(1) for additional information on this command.

The following example sends the test1 file to the system host6:

```
# uucp –r test1 host6! ~/test1
# /usr/lib/uucp/uutry host6
```

10.3 Viewing the Error Log File

To diagnose kernel and hardware errors, you can look at the system events that occurred prior to the errors. Messages from system events, such as error messages relating to the software kernel and system hardware, and informational messages about system status, startup, and diagnostics, are recorded in the binary error log file, `/var/adm/binary.errlog`.

Because this log file is in binary format, the operating system offers special utilities, Compaq Analyze and DECevent, that read the binary log file and run the data through a formatter to display the information. See `ca`(8) and `dia`(8) for more information about Compaq Analyze and DECevent, respectively.

Note that these utilities are not available in the operating system by default; you must install them separately.

Compaq Analyze is part of the Web-Based Enterprise Services (WEBES) kit, a suite of diagnostic utilities that is available for installation from the

Associated Product CD-ROMs. For more information about the WEBES kit, see the following URL:

http://www.compaq.com/support/svctools/webes

DECevent is also available for installation from the Associated Product CD-ROMs, or you can download it from the web. For more information about the DECevent kit, see the following URL:

http://www.compaq.com/support/svctools/decevent

See the *System Administration* manual for information about using the Event Viewer to present errors as interpreted by Compaq Analyze and DECevent. Also, see `uerf`(8) for an alternative to these utilities.

10.4 Viewing the syslogd Daemon Message Files

You can use the `syslogd` daemon to help diagnose session layer problems such as access control problems for the Internet Protocol Version 4 (IPv4) and Internet Protocol Version 6 (IPv6).

The `syslogd` daemon starts running when you boot the system and whenever it receives a hangup signal. By default, it records the system messages for these events in a set of files in the `/var/adm/syslog.dated` directory (as specified in the `/etc/syslog.conf` file). The system messages can indicate error conditions or warnings, depending on the priority codes they contain.

Although it is possible to review the contents of the system message files from the command line, it is best to use the Event Viewer that is part of the SysMan Menu utility, because it simplifies access to the files and makes it easier for you to find particular problems.

To start the Event Viewer, invoke the SysMan Menu as decribed in Section 1.2.1, then select Monitoring and Tuning→View events. Alternatively, you can invoke the Event Viewer from a command line by entering the following command:

```
# /usr/bin/sysman event_viewer
```

Once the Event Viewer is displayed, you can use it to sort the log entries, filter the entries (for a certain event name, priority level, posting host, or date), and obtain more detailed information about individual entries.

For more information about event management and accessing the system log files, see `evm`(5), `syslogd`(8), the *System Administration* manual, and the online help.

11

Testing DNS Servers

In concept, testing DNS/BIND servers consists of locating the information you need. In practice, testing DNS servers involves tracing through a network of servers and their databases to find the server responsible for the information. This chapter describes the tests you can use to locate the information.

11.1 Glossary

The following terms are used in this section. Refer to them as needed during the problem solving tests.

authoritative server

A server that stores information locally. Master and slave servers are examples of authoritative servers. They have primary and secondary authority, respectively, for a given domain.

In contrast, a server that is not authoritative must ask other servers for information about the target host. A forward-only server is an example of this type of server because it forwards queries to a list of forwarders that can answer such requests.

current server

The server you are currently logged in to and running tests from.

data types

The types of resource records in the DNS database files. See `named`(8) for a complete list and explanation.

forwarder

A server that can answer DNS queries from data in its databases and cache, whether or not it is authoritative for the information. Forwarder entries can be in the `named.conf` file.

nameserver (NS) record

Nameserver records map a domain name to a system that serves the domain, and determine whether a system is familiar with the name servers for the authoritative domain. Nameserver records have the following form:

```
domain-name          IN      NS      machine-name
```

On the left is the domain name; on the right is the name of the machine that services the domain.

master server
A server that stores the main copy of a target domain's databases. A master server has primary authority for name service information in a given domain.

slave server
A server that pulls a copy of the target domain's data from another server. In most cases, the data is pulled from a master server. However, in some cases, the data is pulled from another slave server.

A slave server has secondary authority for name service information in a given domain.

start of authority (SOA) record
Start of authority records mark the start of a zone of authority. They occur at the beginning of each master database file. SOA records have the following form:

```
domain-name          IN      SOA      machine-name
```

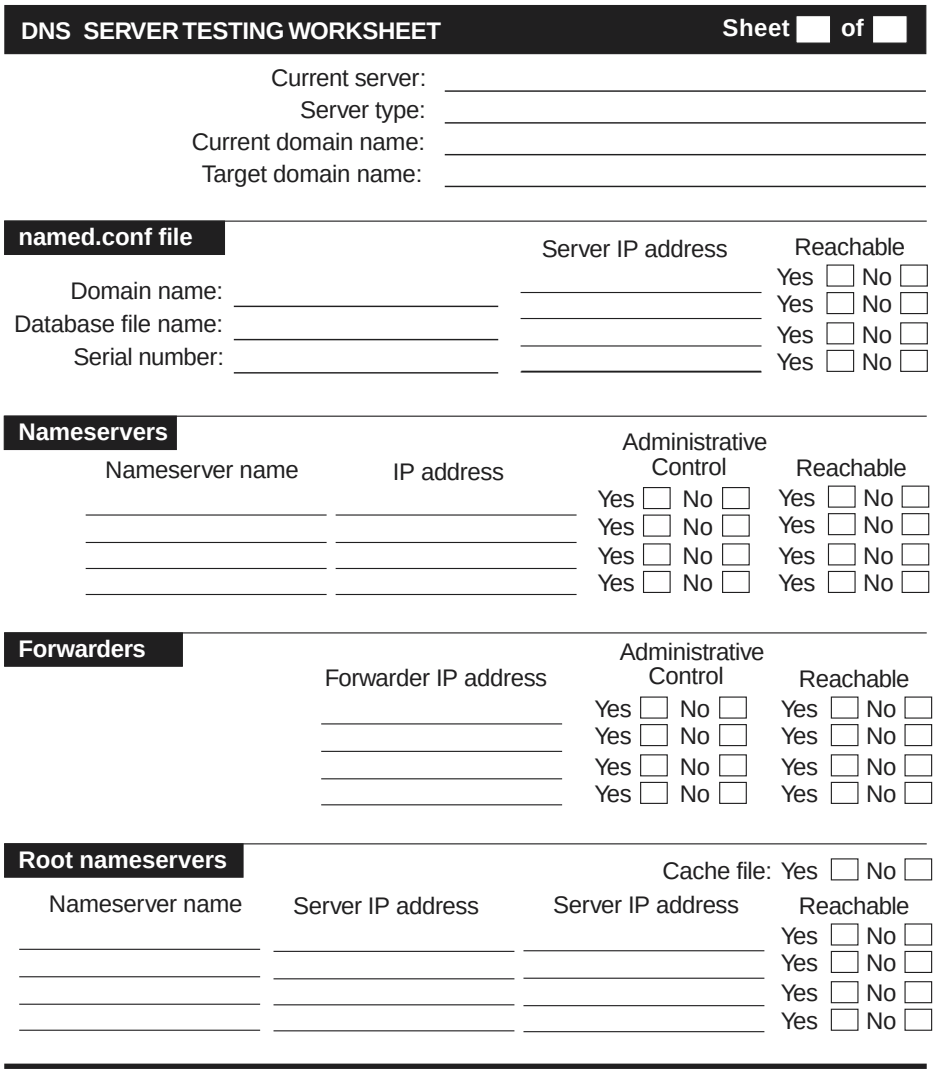
target domain name
The portion of the target host name that begins after the first period (.).

target host
The host name you are trying to resolve. The target domain name is derived from the target host name.

11.2 DNS Server Testing Worksheet

Figure 11–1 shows the DNS Server Testing Worksheet, which you can use to record information from the tests in the following sections. If you are viewing this manual online, you can use the print feature of your browser to print a copy of this worksheet. On a copy of the worksheet, write the current server's name, current domain name, and target domain name.

Figure 11–1: DNS Server Testing Worksheet



DNS SERVER TESTING WORKSHEET Sheet ☐ of ☐

Current server: _____
Server type: _____
Current domain name: _____
Target domain name: _____

named.conf file

		Server IP address	Reachable
Domain name:	_____	_____	Yes ☐ No ☐
Database file name:	_____	_____	Yes ☐ No ☐
Serial number:	_____	_____	Yes ☐ No ☐
		_____	Yes ☐ No ☐

Nameservers

Nameserver name	IP address	Administrative Control	Reachable
_____	_____	Yes ☐ No ☐	Yes ☐ No ☐
_____	_____	Yes ☐ No ☐	Yes ☐ No ☐
_____	_____	Yes ☐ No ☐	Yes ☐ No ☐
_____	_____	Yes ☐ No ☐	Yes ☐ No ☐

Forwarders

Forwarder IP address	Administrative Control	Reachable
_____	Yes ☐ No ☐	Yes ☐ No ☐
_____	Yes ☐ No ☐	Yes ☐ No ☐
_____	Yes ☐ No ☐	Yes ☐ No ☐
_____	Yes ☐ No ☐	Yes ☐ No ☐

Root nameservers

Cache file: Yes ☐ No ☐

Nameserver name	Server IP address	Server IP address	Reachable
_____	_____	_____	Yes ☐ No ☐
_____	_____	_____	Yes ☐ No ☐
_____	_____	_____	Yes ☐ No ☐
_____	_____	_____	Yes ☐ No ☐

11.3 Starting the DNS Server Test

To determine if the current server can resolve the target data, complete the following steps:

1. Determine whether the current server can access the target data. Use the following commands:

```
# nslookup
Default Server: host1.corp.com
Address:  127.0.0.1

> server localhost
Default Server: localhost.corp.com
Address:  127.0.0.1

> set timeout=45
> set retry=2
> target_host.target_domain.
```

If the nslookup command:	Action:
Succeeds	Go to step 3.
Fails	If the first time, go to step 2.
	If the second time, go to Section 11.4.

2. Determine whether the named daemon is running by using the following command:

```
# ps gax | grep named
```

If the named daemon is:	Action:
Running	Go to step 1.
Not running	Start the daemon by using the /sbin/init.d/named start command. If the Internet name service started message is displayed, go to step 1.
	If the message is not displayed, this machine is not configured as a DNS server. Decide how the machine is to be configured. See Section 2.5 for more information.

3. Log in to the client system and use the nslookup command to try to access the target data.

If the `nslookup` command:	Action:
Succeeds	STOP. The client can resolve the target data.
Fails	The server knows the information, but is not transferring it to the client. Log out from the client; restart DNS on the server by using the `/sbin/init.d/named restart` command; log in to the client; and use the `nslookup` command. If it cannot resolve the target data, you have the wrong server or the DNS server is malfunctioning.

11.4 Determining the Server Type

To determine whether the current server is a master server or a slave server, complete the following steps:

1. Compare the target domain name with all domain names of the master and slave entries in the `/etc/named.conf` file. These entries have the following form:

   ```
   zone "domain" {     type server-type;     file
   "filename.db"; };
   ```

   The following example shows the zz.bb.cc. target domain and subsets of this target domain:

   ```
   # cat /etc/named.conf
      .
      .
   options {
           directory "/etc/namedb";
           };

   zone "aa.bb.cc" { 1
           type master;
           file "aa.bb.cc.db";
   };

   zone "cc" { 2
           type master;
           file "cc.db";
   };

   zone "bb.cc" { 3
           type slave;
           file "bb.cc.db";
           masters {
              128.102.0.42;
           };
   };
   ```

```
zone "zz.bb.cc" { [4]
        type slave;
        file "zz.bb.cc.db";
        masters {
           128.102.29.73;
        };
};
   ⋮
```

[1] This zone entry is not a subset of the zz.bb.cc. domain.

[2] This zone entry is a subset of the zz.bb.cc. domain. The server is a master server for the cc. domain and it stores the information for this domain in the `cc.db` file.

[3] This zone entry is a subset of the zz.bb.cc. domain. The server is a slave server for the bb.cc. domain and it stores the information for this domain in the `bb.cc.db` file.

[4] This zone entry matches the zz.bb.cc. domain. The server is a slave server for the zz.bb.cc. domain and it stores the information for this domain in the `zz.bb.cc.db` file.

For more information on the format of the `named.conf` file, see `named.conf`(8) and the *BIND Configuration File Guide*.

When directed, record information in the `named.conf` file section on the worksheet.

If a `named.conf` entry:	And the type is:	Action:
Matches the target domain name	Master	Write the server type, domain name, and database file name on the worksheet and go to Section 11.8.
	Slave	Write the server type, domain name, database file name, and host IP addresses on the worksheet and go to Section 11.7.
Is a subset of the target domain name	Master	Write the server type, domain name, and database file name on the worksheet and go to step 2.
	Slave	Write the server type, domain name, database file name, and host IP addresses on the worksheet and go to step 2.
Neither matches nor is a subset of the target domain name	Master or slave	Go to Section 11.5.

2. Compare the target domain name with all nameserver (NS) records in the database file recorded on the worksheet. When directed, record information in the Nameservers section on the worksheet. Use the following commands to create and view a list of NS records:

```
# grep –n NS database_file > ns_list
# grep –n ORIGIN database_file >> ns_list
# sort –n ns_list > ns_list.srt
# cat ns_list.srt
```

The following example shows the file created by these commands. The target domain is zz.bb.cc.:

```
# cat ns_list.srt
1:$ORIGIN cc.
10:             IN          NS          server_1.cc.
17:$ORIGIN cc.
18:bb           IN          NS          server_3.bb.cc.
21:$ORIGIN cc.
22:bb           IN          NS          server_4.bb.cc.
41:$ORIGIN bb.cc.
42:zz           IN          NS          server_5.zz.bb.cc. [1]
45:$ORIGIN bb.cc.
46:zz           IN          NS          server_6.bb.cc. [2]
```

[1] This entry is a longer subset (exact match) of the target domain. The domain name from the preceding $ORIGIN line, .bb.cc., is appended to the domain name of this line, zz, resulting in zz.bb.cc..

[2] This entry is a longer subset (exact match) of the target domain. The domain name from the preceding $ORIGIN line, .bb.cc., is appended to the domain name of this line zz, resulting in zz.bb.cc..

If any NS record:	And the server is:	Action:
Contains a longer subset of the target domain name than the domain name on the worksheet	Master or slave	The server has neither primary nor secondary authority for the target information. Write the names of the servers on the worksheet and go to step 3.
Does not contain a longer subset of the target domain name than the domain name on the worksheet	Master	The database files contain the target information. Go to Section 11.8.
	Slave	The database files contain the target information. Go to Section 11.7.

3. Find the IP addresses in the database file for any name servers on the worksheet. Use the following commands:

```
# grep –n ORIGIN database_file > ip_list
# grep –n server_name database_file >> ip_list
   ⋮
# sort –n ip_list > ip_list.srt
# cat ip_list.srt
```

Write the IP addresses on the worksheet next to the corresponding server name and go to Section 11.5. The following example shows the file created by the preceding commands:

```
# cat ip_list.srt
1:$ORIGIN cc.
17:$ORIGIN cc.
21:$ORIGIN cc.
41:$ORIGIN bb.cc.
42:zz           IN         NS        server_5.zz.bb.cc.
43:$ORIGIN zz.bb.cc.
44:server_5     IN         A         10.140.48.3 [1]
45:$ORIGIN bb.cc.
46:zz           IN         NS        server_6.bb.cc.
47:$ORIGIN bb.cc.
48:server_6     IN         A         10.12.48.3 [2]
```

[1] The IP address for server_5 is 10.140.48.3.

[2] The IP address for server_6 is 10.12.48.3.

11.5 Finding the Target Domain Information

To determine the servers that the current server communicates with in order to get information for the target domain, complete the following steps:

1. Search the `named.conf` file and find any forwarder entries. These entries have the following form:

```
options {
        directory "directory-name";
        forward only;
        forwarders {
                IP-address;
                IP-address;
        };
};
```

When directed, record information in the Forwarders section on the worksheet.

If your system:	Action:
Contains a forwarder line	The current server forwards requests. Write the IP addresses for any forwarders on the worksheet and go to Section 11.6.
Does not contain a forwarder line	The current server does not forward queries. Go to step 2.

2. Compare the target domain name with all nameserver (NS) records in the database file recorded on the worksheet. When directed, record information in the Nameserver section on the worksheet.

   Use the following commands to create and view a list of NS records for each database file:

```
# grep –n NS database_file > ns_list
# grep –n ORIGIN database_file >> ns_list
# sort –n ns_list > ns_list.srt
# cat ns_list.srt
```

If any NS record:	And:	Action:
Contains a longer subset of the target domain name than the domain name on the worksheet	→	Write the names of the servers on the worksheet and go to step 3.
Does not contain a longer subset of the target domain name than the domain name on the worksheet	The Nameserver section on the worksheet is blank	Go to Section 11.9.

3. Find the IP addresses in the database file for any name servers on the worksheet. Use the following commands:

```
# grep –n ORIGIN database_file > ip_list
# grep –n server_name database_file >> ip_list
   ⋮
# sort –n ip_list > ip_list.srt
# cat ip_list.srt
```

   Write the IP addresses on the worksheet next to the corresponding server name and go to step 4.

4. Verify whether each server listed in the Nameserver section on the worksheet is reachable by using the `ping` command.

If a server:	And:	Action:
Responds to the `ping` command	You have root access to the server	The server is reachable and under your administrative control. Note both items on the worksheet. Go to step 5.
	You do not have root access to the server	The server is reachable, but not under your administrative control. Note both items on the worksheet. Go to step 5.
Does not respond to the `ping` command	→	Note this on the worksheet.
		If no servers responded to the `ping` command, STOP. The current server is isolated from its servers on the network. You cannot solve the problem; contact your enterprise network administrator.

5. Log in to each reachable server by using the `telnet` command. Each server you log in to becomes the current server. Get a new worksheet and write the current server name, current domain name, and target domain name on it. For each server, perform the DNS server test. See Section 11.3.

11.6 Testing the Forwarders

To determine whether the forwarders prevent you from resolving the target host name, complete the following steps:

1. Determine whether each forwarder listed on the worksheet is reachable by using the `ping` command.

If a forwarder:	And:	Action:
Responds to the `ping` command	You have root access to the forwarder	The forwarder is reachable and under your administrative control. Note both items on the worksheet. Go to step 2.
	You do not have root access to the forwarder	The forwarder is reachable, but not under your administrative control. Note both items on the worksheet. Go to step 2.

If a forwarder:	And:	Action:
Does not respond to the `ping` command	→	Note this on the worksheet.
		If no forwarders responded to the `ping` command, STOP. The current server is isolated from its forwarders on the network. You cannot solve the problem; contact your enterprise network administrator.

2. Edit the `named.conf` file and eliminate any forwarders that did not respond to the `ping` command.
3. Enter the `nslookup` command again for the target host.

If the `nslookup` command:	Action:
Succeeds	Go to step 4.
Fails	Go to step 5.

4. Edit the `named.conf` file and add the forwarders removed in step 2 at the end of the forwarders line. In addition, contact the administrators of forwarders not under your administrative control and inform them that they might have a problem with their forwarder. STOP.
5. Log in to each reachable forwarder by using the `telnet` command. This forwarder is now the current server. On a new worksheet, write the current server name, current domain name, and target domain name. For each server, perform the DNS server test. See Section 11.3.

If the forwarder or other machines:	Action:
Cannot resolve the target name	Remove the forwarder from `named.conf` file.
Can resolve the target name	STOP.

11.7 Testing Slave Servers

To determine whether the slave server contains the target data, complete the following steps:

1. Find the database serial number in the start of authority record in the database file. Use the following command:

```
# head –4 database_file
```

Write the first number, which is the serial number, on the worksheet in the `named.conf` section. If you have a serial number on a previous worksheet, compare the current serial number with that one. Note whether the current number is larger (newer) or smaller (older) than the other number. In the following example, 23 is the serial number:

```
# head –4 database_file
$ORIGIN cc.
bb          IN        SOA        host1.bb.cc. postmaster.host1.bb.cc. (
            23 300 60 1209600 43200 )
            IN        MX         100 host1.bb.cc.
```

2. Determine whether the target data is contained in the database file written on the worksheet. Use the following commands to create and view a list of resource records:

```
# grep –n data_type database_file > ns_list
# grep –n ORIGIN database_file >> ns_list
# sort –n ns_list > ns_list.srt
# cat ns_list.srt
```

If the database file:	And the serial number is:	Action:
Contains the target data	Newer	The data exists in the domain. Go to step 3.
Contains the target data	Older or same	The server is malfunctioning or you made a error. Verify all steps up to this point.
Does not contain the target data	→	The data does not exist in the domain. Go to step 4.

3. Determine whether the current server can access the target data. Use the following commands:

```
# nslookup
Default Server: host1.corp.com
Address:  127.0.0.1

> server localhost
Default Server: localhost.corp.com
Address:  127.0.0.1

> set timeout=45
> set retry=2
> target_host.target_domain.
```

If the `nslookup` command:	And the database serial number is:	Action:
Succeeds	→	STOP. The server is working. Either the client or server cannot communicate with the server or this server just started working.
Succeeds	Newer	Log out of the slave server. Get the previous slave server's worksheet and go to step 8.
Fails	→	Restart the current slave server by using the `/sbin/init.d/named` restart command. Then reenter the `nslookup` command.

4. Verify whether each name server listed on the worksheet is reachable by using the `ping` command.

If a server:	And:	Action:
Responds to the `ping` command	You have root access to the server	The server is reachable and under your administrative control. Note both items on the worksheet.
	You do not have root access to the server	The server is reachable, but not under your administrative control. Note both items on the worksheet.
Does not respond to the `ping` command	→	Note this on the worksheet.
		If no servers responded to the `ping` command, STOP. The current server is isolated from its servers on the network. You cannot solve the problem; contact your enterprise network administrator.

Count the number of servers that responded to the `ping` command and that are under your administrative control. If the number is zero (0), go to Section 11.10.

5. Edit the `named.conf` file and find the `slave` entry. Delete the IP address for those servers that are not reachable and are not under your administrative control. Delete those entries from the worksheet as well.
6. Log in to each reachable server by using the `telnet` command. Start a new worksheet for each server, writing the server name as the current server. Save the old worksheet.
7. Compare the target domain name with all domain names of the master and slave entries in the `/etc/named.conf` file. These entries have the following form:

```
zone "domain" {
    type server-type;
    file "filename.db";
};
```

When directed, record information in the `named.conf` file section on the worksheet.

If a `named.conf` entry:	And the first field is:	Action:
Matches the target domain name	Master	Write the domain name and database file name on the worksheet and go to Section 11.8.
	Slave	Write the domain name, host IP addresses, and the database file name on the worksheet and go to step 1.
Is a subset of the target domain name	→	STOP. Examine another master or slave server entry.
Neither matches nor is a subset of the target domain name	→	STOP. Examine the next master or slave server entry.

8. Restart the current slave server by using the following command:

```
# /sbin/init.d/named restart
```

After restarting, wait a few minutes before proceeding to the next step. This allows time for the database to be updated.

9. Determine whether the current server can access the target data. Use the following commands:

```
# nslookup
Default Server: host1.corp.com
Address:  127.0.0.1

> server localhost
```

```
Default Server: localhost.corp.com
Address:  127.0.0.1

> set timeout=45
> set retry=2
> target_host.target_domain.
```

If the nslookup command:	Action:
Succeeds	STOP. The server is working. If you are in a `telnet` session to another slave server, log out and go to step 8.
Fails	If you just ended a `telnet` session to another server, go to step 10. If you did not end a `telnet` session, either the current server is malfunctioning and cannot read the database file or you made an error. Verify all steps up to this point.

10. Compare the database serial number of the current server with the database serial number of the server from which you just logged out. Use the following command:

    # **head –4** ***database_file***

If the current database serial number is:	Action:
Older	Either the server cannot pull the database from the authoritative server or you made an error. Verify all steps up to this point.
The same	The serial numbers cannot be equal. Verify all steps up to this point.

11.8 Testing Master Servers

To determine whether the master server contains the target data, complete the following steps:

1. If you are in a `telnet` session from a slave server to a master server, go to step 2. Otherwise, go to step 3.
2. Find the database serial number in the start of authority record in the database file. Use the following command:

   # **head –4** ***database_file***

   Write the first number, which is the serial number, on the worksheet in the `named.conf` section. If you have a serial number on a previous worksheet, compare the current serial number with that one. Note

whether the current number is larger (newer) or smaller (older) than the other number. In the following example, 23 is the serial number:

```
# head –4 database_file
$ORIGIN cc.
bb          IN        SOA        host1.bb.cc. postmaster.host1.bb.cc. (
            23 300 60 1209600 43200 )
            IN        MX         100 host1.bb.cc.
```

3. Determine whether the target data is contained in the database file written on the worksheet. Use the following commands to create and view a list of resource records:

```
# grep –n data_type database_file > ns_list
# grep –n ORIGIN database_file >> ns_list
# sort –n ns_list > ns_list.srt
# cat ns_list.srt
```

If the database file:	Action:
Contains the target data	The data exists in the domain. Go to step 4.
Does not contain the target data	The data does not exist in the domain. Go to step 5.

4. Determine whether the current server can access the target data. Use the following commands:

```
# nslookup
Default Server: host1.corp.com
Address:  127.0.0.1

> server localhost
Default Server: localhost.corp.com
Address:  127.0.0.1

> set timeout=45
> set retry=2
> target_host.target_domain.
```

If the nslookup command:	And the database serial number is:	Action:
Succeeds	→	STOP. The server is working. Either the last server cannot communicate with this server or this server just started working.
Succeeds	Older or same	STOP. The server is malfunctioning or you made an error. Verify all steps up to this point.

If the nslookup command:	And the database serial number is:	Action:
Succeeds	Newer	Log out of the master server. Get the previous slave server's worksheet and go to step 8 in Section 11.7.
Fails	→	Restart the current master server by using the /sbin/init.d/named restart command. Then reenter the nslookup command.

5. Edit the database file and increment the database serial number by 1 to age the database. The following example shows the SOA record before and after editing. Note the serial number increase from 23 to 24.

```
# head –4 database_file
$ORIGIN cc.
bb          IN        SOA        host1.bb.cc. postmaster.host1.bb.cc. (
            23 300 60 1209600 43200 )
            IN        MX         100 host1.bb.cc.
# vi database_file
   .
   .
   .
# head –4 database_file
$ORIGIN cc.
bb          IN        SOA        host1.bb.cc. postmaster.host1.bb.cc. (
            24 300 60 1209600 43200 )
            IN        MX         100 host1.bb.cc.
```

6. Edit the database file and add new data to the database. Refer to Section 11.1 for information on valid data types. Precede any new entry with a $ORIGIN entry, and separate database fields with a tab character. The following example shows a new address record for host host1.bb.cc.:

```
$ORIGIN bb.cc
host1             IN         A            16.141.112.11
```

7. Restart the master server by using the following command:

```
# /sbin/init.d/named restart
```

8. Determine whether the current server can access the target data. Use the following commands:

```
# nslookup
Default Server: host1.corp.com
Address:  127.0.0.1

> server localhost
Default Server: localhost.corp.com
Address:  127.0.0.1

> set timeout=45
> set retry=2
> target_host.target_domain.
```

If the `nslookup` command:	Action:
Succeeds	Log out of the master server. Get the previous slave server's worksheet and go to step 8 in Section 11.7.
Fails	Either the server is malfunctioning or you made an error. Verify all steps up to this point.

11.9 Tracing Information from the Root Name Server

To resolve the target name beginning with the root of the DNS namespace, complete the following steps:

1. Determine whether the current server has a cache file containing the information necessary to find a root server. Use the following command:

   ```
   # grep cache /etc/named.conf
   ```

If a cache line:	Action:
Does not exist	The current server cannot contact a root name server. Note this on the worksheet and go to step 2.
Exists	Note this on the worksheet and go to step 3.

2. Add a cache file to your server.

   Caution

   Adding a cache file alters many system files. Perform the following steps as shown to ensure the correct operation of your system.

   a. Create copies of specific DNS and system files. Enter the following commands:

   ```
   # cd /etc
   # cp –r namedb namedb.back
   # cp rc.config.common rc.config.common.back
   # cp hosts hosts.back
   # cp resolv.conf resolv.conf.back
   # cp svc.conf svc.conf.back
   # cd /var/adm/sendmail
   # cp sendmail.cf sendmail.cf.back
   ```

   b. Display the name of the local host by using the `hostname` command. You will need to reset the host name after running the SysMan Menu utility and copying system files.

c. Run the SysMan Menu utility. Modify the configuration and create a caching server (see Section 2.5.3). Do not start the DNS daemon automatically and do not run `svcsetup`.

d. Copy the system files to the `/etc` directory. Use the following commands:

```
# cd /etc
# cp rc.config.common.back rc.config.common
# cp hosts.back hosts
# cp resolv.conf.back resolv.conf
# cp svc.conf.back svc.conf
```

e. Set the host name to the original host name by using the `hostname` command.

f. Copy the `sendmail` file to the `/var/adm/sendmail` directory and restart `sendmail`. Use the following commands:

```
# cd /var/adm/sendmail
# cp sendmail.cf.back sendmail.cf
# /sbin/init.d/sendmail restart
```

g. Copy the DNS files to the `/etc` directory. Use the following commands:

```
# cd /etc
# cp namedb/namedb.boot namedb.back/named.conf_new
# cp namedb/namedb.ca namedb.back
# rm –rf namedb.back namedb
# mv namedb.back namedb
# cd namedb
```

h. Edit the `named.conf` file and add the following lines to the end of the file:

```
zone "." {
   type hint;
   file "named.ca";
};
```

i. Remove the `named.conf_new` file.

j. Restart the current server by using the `/sbin/init.d/named restart` command.

3. Display the `named.ca` file by using the following command:

```
# cat named.ca
```

Write the root name server names and IP addresses in the Root nameservers section on the worksheet.

4. Verify whether each root name server listed on the worksheet is reachable by using the `ping` command.

If a root name server:	Action:
Responds to the `ping` command	Note this on the worksheet. Go to Section 11.11.
Does not respond to the `ping` command	Note this on the worksheet. If no servers responded to the `ping` command, go to step 5.

5. Do either of the following:
   - Give the current server access to the Internet. Then restart the `named` daemon by using the following command:

     ```
     # /sbin/init.d/named restart
     ```

     Keep the current server and worksheet, and go to Section 11.3.
   - Add a forwarder entry to direct the current server to communicate with a machine with Internet access. Then restart the `named` daemon by using the following command:

     ```
     # /sbin/init.d/named restart
     ```

     Keep the current server and worksheet, and go to Section 11.3.

11.10 Resolving Target Data

To resolve target data using a name server, complete the following steps:

1. Enter the `nslookup` command for the target system. Choose the first name server from either the Root nameserver section or the Nameserver section. Use the following commands:

   ```
   current_server> nslookup
   Default Server:  localhost.omni.corp.com
   Address:  127.0.0.1

   > server IP_address
   Default Server:  [IP_address]
   Address:  128.102.16.10

   > set type data_type
   > target_name
   ```

If the `nslookup` command:	And:	Action:
Succeeds	→	STOP. The server is working. Either the last server you tested does not talk to this one or this server just started working. Verify all steps completed up to this point.
Fails	An error message is returned.	If a `non-existent domain` message is displayed, no data exists for the *target_name*. Go to Section 11.11. If a `no information available` message is displayed, the *target_name* exists, but is not associated with the target data. If the data is required, contact the target domain administrator and request that the data be added to the domain. If a `timed-out` message is displayed, the server to which you sent the query cannot contact the server that is responsible for the target data. Go to step 2.
Fails	An error message is not returned.	An unknown error. Contact the target domain administrator.

2. Modify the retry and timeout values and re-enter the `nslookup` command. Enter the following commands:

```
current_server> nslookup
Default Server:  localhost.omni.corp.com
Address:  127.0.0.1

> server IP_address
Default Server:  [IP_address]
Address: IP_address

> set type data_type
> target_name
```

If the `nslookup` command:	And:	Action:
Succeeds	→	STOP. The server is working, but is slow. This might prevent the query from being resolved. If the network connection to the server is correct, wait two or three hours for the performance to improve. If it does not improve, contact the server administrator.
Fails	An error message is returned	If a `non-existent domain` message is displayed, no data exists for the *target_name*. Go to Section 11.11. If a `no information available` message is displayed, the *target_name* exists, but the target data is not associated with it. If the data is required, contact the target domain administrator and request that the data be added to the domain. If a `timed-out` message is displayed, the server to which you sent the query cannot access the server that is responsible for the data. Select another nameserver from the worksheet and go to step 1.
Fails	An error message is not returned	An unknown error. Contact the target domain administrator.

11.11 Finding the First Nonexistent Domain

To find the first nonexistent domain in a target name, complete the following steps:

1. Enter the `nslookup` command, using the smallest subset of the target domain name. Enter the following commands:

```
current_server> nslookup
Default Server:  localhost.omni.corp.com
Address:  127.0.0.1

> server IP_address
Default Server:  [IP_address]
Address: IP_address

> set type=ns
```

```
> target_name_subset
```

For example, if the target domain name is zz.bb.cc., the first attempt is to resolve the target name subset cc.. If necessary, the second attempt uses bb.cc., and the third, zz.bb.cc..

If the `nslookup` command:	And:	Action:
Succeeds	→	Go to step 3.
Fails	An error message is returned	If a `non-existent domain` message is displayed, no data exists for the *target_name*. If the data is required, contact the domain administrator and request that the data be added to the domain. STOP. If a `timed-out` message is displayed, go to step 2.

2. Modify the retry and timeout values and enter the `nslookup` command again. Enter the following commands:

```
current_server> nslookup
Default Server:  localhost.omni.corp.com
Address:  127.0.0.1

> server IP_address
Default Server:  [IP_address]
Address: IP_address

> set retry=2
> set timeout=45
> set type=ns
> target_name_subset
```

If the `nslookup` command:	And:	Action:
Succeeds	→	Go to step 3.
Fails	An error message is returned	If a `non-existent domain` message is displayed, no data exists for the *target_name*. If the data should exist, contact the domain administrator and request that the data be added to the domain. STOP. If a `timed-out` message is displayed, select another name server from the worksheet and go to Section 11.10.

3. Add the next part of the target domain name to the target subset and go to step 1.

12

Reporting Network Problems

If you are unable to solve a critical problem with the network or network service, do the following:

1. Read the release notes for the product to see if the problem is known. If it is, follow the solution offered to solve the problem.
2. Determine whether the product is still under warranty or whether your company purchased support services for the product. Your operations manager can supply you with the necessary information.
3. If either condition in step 2 was met, take one of the following actions:

   a. Access the online service database, if you have purchased this service, and determine if the problem you are experiencing has already been reported. If it has not, log your problem.

   b. Call your service representative and describe the problem.

4. If you are requested to supply any information pertaining to the problem, gather the necessary information and submit it.

You might be asked to submit some information that can help isolate problems to a particular area of the system and speed the resolution of the problem. It is a good idea to keep all basic information in a `system.information` file. Then you can easily include it with your problem report.

The following sections describe some of the information that you might be asked to submit.

12.1 Gathering General Information

Gather the following information about your system:

- The operating system version and revision number (from the `/etc/motd` file). Add this to the `system.information` file.
- A description of your system's activity before the error.
- A listing of the exact command line or lines executed and the output.

- A copy of the application source code, if running a user-created application. If possible, include a sample test program that demonstrates the problem.

12.2 Gathering Hardware Architecture Information

Gather the following information about the hardware architecture:

- A description of the model of the workstation or server (from the `/usr/sys/conf/`*`HOSTNAME`* file), including the type of graphics controller (if a workstation), the amount of memory, and third-party hardware
- A description of the X server

  To determine which type you are running, enter the following command:

  ```
  # ps ax | grep /usr/bin/X >> system.information
  ```

- A description of the disks used and the size of your swap partition

  For example, if your system disk is unit 0, enter the following commands as root to add this information to the `system.information` file:

  ```
  # disklabel –r /dev/rrz0a >> system.information
  # echo df: >> /system.information
  # df >> /system.information
  # echo mount: >> /system.information
  # mount >> /system.information
  # echo xdpyinfo: >> /system.information
  # xdpyinfo >> /system.information
  ```

- Any networking information

  To add this to the `system.information` file, enter the following commands:

  ```
  # echo netstat: >> /system.information
  # netstat –i –n >> system.information
  # netstat –r –n >> /system.information
  # echo nslookup: >> /system.information
  # nslookup localhost >> /system.information
  ```

- Any event logging information

  To add this to the `system.information` file, enter the following commands:

  ```
  # uerf –R –o full | head –200 >> /system.information
  ```

12.3 Gathering Software Architecture Information

Gather the following information about the software architecture:

- A description of the software subsets installed

To add this to the system.information file, enter the following commands:

```
# echo setld: >> /system.information
# setld –i >> /system.information
```

- The output of the setld log file

  To add this to the system.information file, enter the following command:

  ```
  # pr /usr/adm/smlogs/setld.log >> /system.information
  ```

- The automatic reboot file

  To add this to the system.information file, enter the following commands:

  ```
  # pr /etc/rc.config* >> /system.information
  # pr /sbin/rc[023] >> /system.information
  # pr /sbin/init./* >> /system.information
  ```

- A description of the layered products installed

A

Writing Automount and AutoFS Maps

This appendix explains how to write AutoFS and Automount maps, which you can use to specify the file systems you intend to mount with the `automount` daemon or `autofsd` daemon (see Section 4.1.2 and Section 4.6.3).

It also describes some fundamental differences between Automount and AutoFS, particularly in how these services interpret and execute the specified mounts.

Note

The Automount daemon will be retired in a future release of the operating system. For information about migrating from Automount to AutoFS, see Section 4.6.3.5.

With a few restrictions, as documented in this appendix and in the Restrictions section of `autofsmount`(8), Automount and AutoFS maps can be used interchangeably.

A.1 Map Conventions and Basic Syntax

Automount and AutoFS maps are configuration files that indicate which remote file systems to auto-mount, where to mount them, and which mount options to use. They can also contain entries that point to other map files in a hierarchical manner.

Conventionally, these map files are named with the prefix `auto` and they are located in the `/etc` directory on the local system. However, you can also distribute maps via NIS. Any system running Automount or AutoFS can use local maps, NIS maps, or a combination of both. See Section 4.6.3.4 for more information.

There are four types of maps:

- Master map
- Direct map
- Indirect map
- Special map

The following sections explain the purpose and syntax for each type of map.

A.1.1 Master Map

The master map is the top-level map file for Automount or AutoFS. It contains only entries that point to other maps (direct, indirect, or special); it does not describe any mounts. However, it can contain mount options that apply to all of the mounts listed in the specified maps.

Each line in the master map has the following syntax:

key map [*mount-options*]

key

Names the directory for which the specified map applies. If the *map* argument is the name of an indirect map or the name of a special map, enter the full pathname of a local directory. If the map argument is a direct map, use the predefined dummy directory, `/-`. See Section A.1.2 and Section A.1.3 for more information.

map

Names the map that the `automount` or `autofsmount` command uses to find the mount points and locations. This can be a file name, an NIS map name, or a special map name.

mount-options

Lists the options used to regulate the mounting of entries listed in *map*.

The following map is an example of a master map:

```
/-          auto.tools    -nosuid,hard,intr  1
/home       auto.home     -nosuid,hard,intr  2
/net        -hosts        -nosuid,hard,intr  3
```

In this master map, the entries have the following purpose:

1. Points to a direct map called `auto.tools`, which describes mounts for shared applications.
2. Points to an indirect map called `auto.home`, which describes mounts for users' home directories.
3. Points to a special map called `-hosts`, which is described in Section A.1.4.

All of the entries include a string of mount options that apply to all mounts described within the respective map files. For a description of these mount options, see `mount`(8).

A.1.2 Direct Map

Direct maps specify the remote file systems to mount locally, the remote hosts that are serving those file systems, and the local directories on which those file systems are to be mounted. You can also use direct maps to specify the mount options for each file system.

Direct maps are so named because each mount they describe is associated with a fully qualfied mount point. (In contrast, the entries in indirect maps are associated with a parent directory that is specified in the master map.)

Direct maps have the following syntax:

key [*mount-options*] *location*

key

Specifies the full pathname of the mount point.

mount-options

Lists the options for this specific mount. When present, these options override any mount options specified on the command line or in the master map.

location

Specifies the location of the resource being mounted and uses the format *server:pathname*. (For the mount to be successful, the file system must be exported from the specified server to the local host as described in Section 4.5.2.) Multiple *location* fields can be specified; see Section A.2.5 for more information.

The following map, `auto.tools`, is an example of an direct map:

```
/tools/bin   apollo:/usr/opt/bin
/tools/lib   apollo:/usr/opt/lib
/tools/man   apollo:/usr/opt/man
```

It describes the mounts for shared applications that are exported from the apollo production server. On that server, the file systems are located in the `/usr/opt` directory; however, on the client system, they are mounted in the `/tools` directory.

A.1.3 Indirect Map

Indirect maps have the same purpose and format as direct maps. However, unlike the key in a direct map, the key in an indirect map is a simple directory name that does not begin with a slash.

An indirect map as a whole is associated with a parent directory that is specified in the master map (as in Section A.1.1) or on the command line. The entries in an indirect map list subdirectories that are individually mounted within that parent directory.

The following map, `auto.home`, is an example of an indirect map:

```
strauss   apollo:/usr/staff/strauss
cameron   apollo:/usr/staff/cameron
smith     zeus:/usr/staff/smith
samler    apollo:/usr/staff/samler
campbell  zeus:/usr/staff/campbell
larson    apollo:/usr/staff/larson
```

This map describes the mounts for users' home directories, which are exported from two production servers, zeus and apollo.

Because all of the user's directories listed in this map are mounted in the same parent directory, `/home`, there is no need to specify it in each entry. Instead, the indirect map as a whole is associated with the `/home` directory.

For example, the first entry in this file would be logically equivalent to the following direct map entry:

```
/home/strauss   apollo:/usr/staff/strauss
```

A.1.4 Special Maps

The `-hosts` map and the `-null` map are special maps that are built into the automatic mounting daemons.

You can use the `-hosts` map to simultaneously mount all of the file systems that are exported from an NFS server listed in the local system's `hosts` database. (The `hosts` database that your system uses is determined by the services running on your system and the order in which those services are specified in the `/etc/svc.conf` file.)

You can use the `-hosts` map with the `automount` command as follows:

```
# automount /net -hosts
```

Similarly, you can use the `-hosts` map with the `autofsmount` command, as follows:

```
# autofsmount /net -hosts
```

Or, you can use the `-hosts` map in a map file, as shown in Section A.1.1.

If a user on the local system subsequently switches into the `/net/`*`hostname`* directory, where *`hostname`* is a server listed in the `hosts` database, all of the exported file systems from *`hostname`* are automatically mounted on the local system under the `/net/`*`hostname`* directory.

For example, suppose that hera and sheba are both hosts on a local area network that is running NIS. If superuser on hera enters the `automount /net –hosts` command, users on hera can access any directories that sheba exports to hera. All of the exported directories are mounted under `/net/sheba` on hera.

The `-null` map cancels the map associated with the directory indicated. You can use it to cancel a map specified in the master map. For example, invoking the `automount` command in the following manner causes the `/net` entry in `auto.master` to be ignored:

```
# automount /net –null
```

The `-null` map works similarly with the `autofsmount` command, except if you intend to cancel an entry in an indirect map. In this special case, you need to use the `-null` option with the `autofsd` command. For example:

```
# autofsd /works/dorado –null
```

When this command is executed, AutoFS cancels the mount associated with the `/works/dorado` mount point, which is specified in an indirect map.

A.2 Advanced Map Syntax

Automount and AutoFS provide additional syntax that allows you to write less redundant maps, and, in some cases, provides more control over how and when file systems are mounted. This syntax includes:

- Substitution and pattern matching
- Environment variables
- Multiple mounts
- Shared mounts
- Replicated file systems

The following sections describe this syntax in more detail.

A.2.1 Substitution and Pattern Matching

Both the `automount` command and the `autofsmount` command recognize the ampersand (&) and asterisk (*) characters, which allow you to eliminate redundancy within maps.

You can use an ampersand as a substitute for the key name of a map entry whenever that key is repeated in the same map entry.

Ampersands are allowed in both direct and indirect maps; however, they are most efficient and easily understood in the context of indirect maps. The following example is an indirect map that does not use ampersands:

```
#key                mount-options       location
#
host1               –rw,nosuid          host1:/home/host1
host2               –rw,nosuid          host2:/home/host2
```

Using the ampersand (&) as a substitution character, the entries read as follows:

```
#key                mount-options       location
#
host1               –rw,nosuid          &:/home/&
host2               –rw,nosuid          &:/home/&
```

You can use the asterisk (*) to substitute for lines that are all formatted similarly. Both daemons use the asterisk to match any host not listed as a key in an entry before the asterisk.

The asterisk is allowed only in indirect maps. The following example shows how the asterisk (*) is typically used:

```
#key                mount-options       location
#
host1               –rw,nosuid          &:/home/&
host2               –rw,nosuid          &:/home/&
*                   –rw,nosuid          &:/home/&
```

Suppose a user enters the following command:

```
% ls /home/host5
```

Either daemon recognizes the host name, host5, as the `key` and substitutes host5 for each of the ampersands in the `location` field. The map is interpreted as follows for host5:

```
#key                mount-options       location
#
host5               –rw,nosuid          host5:/home/host5
```

Note

The `automount` and `autofsmount` commands ignore any entry that follows an asterisk in a local map file.

A.2.2 Environment Variables

You can use the value of an environment variable in a map by adding a dollar sign ($) prefix to its name. You also can use braces ({ }) to delimit the name of the variable from appended letters or digits.

Environment variables can be inherited from the environment or can be defined explicitly with the -D option on the command line. For example, you can invoke the automount daemon with the HOST variable by entering the following command:

```
# automount –D HOST=hostname
```

To define a variable with AutoFS, you must pass it to both the autofsd daemon and the autofsmount command, as follows:

```
# autofsd –D HOST=hostname
# autofsmount -D HOST=hostname
```

Although you can define new variables while either service is running, you typically define them when you start Automount or AutoFS on your system.

The following is an example of a direct map entry that uses the standard environment variable HOST to specify a particular directory for the local host to mount:

```
/mydir     –rw        apollo:/export/$HOST
```

The following is an example of a direct map entry that uses the explicitly defined environment variables MACH and OS to specify the correct tools directory for the local host's architecture and operating system:

```
/tools     –rw     zeus:/tools/${MACH}.${OS}
```

The following is an example of a direct map entry that uses the explicitly defined environment variable NET to specify the appropriate host name for a server:

```
/share/orchard/build/set5     –rw    {$NET}orchard:/share/orchard/build/set5
```

This is useful if the server is connected to several subnetworks, each referring to the server by a different host name.

A.2.3 Multiple Mounts

When you write direct and indirect maps, you can specify that different directories within a file system hierarchy be mounted from different servers. For example, if you mount the /usr/local file system on your machine, you can mount the various subdirectories within /usr/local from different servers.

The following example shows an entry in a direct map in which the directories /usr/local/bin, /usr/local/src, and /usr/local/man are mounted from the machines host1, host2, and host3, respectively:

```
/usr/local\
                /bin    –ro        host1:/usr/local/bin \
                /src    –ro        host2:/usr/local/src \
                /man  –ro          host3:/usr/local/man
```

This entry, which is displayed over four lines with continuation marks (\) and indentation for readability, is an example of multiple, nonhierarchical mounts. These mounts are nonhierarchical because they are triggered on an individual basis as users switch into the respective directories, despite the fact that all of the mount points are located in the same /usr/local directory.

In contrast, when file systems are mounted hierarchically, the entire hierarchy is treated as one object. When a subdirectory within the hierarchy is referenced, the daemon mounts the entire hierarchy. The entire hierarchy is also unmounted as one object.

The following example shows a true hierarchical entry:

```
/usr/local \
            /           -ro       host0:/usr/local \
            /bin        -ro       host1:/usr/local/bin \
            /src        -ro       host2:/usr/local/src \
            /tools      -ro       host3:/usr/local/tools
```

Here, the administrator adds the mount point / to mount /usr/local from host0, which completes the hierarchy. As a result, when a user switches into any subdirectory within /usr/local, such as /usr/local/bin, the daemon simultaneously mounts the entire /usr/local hierarchy.

The only exception with respect to hierarchical mounts is specific to AutoFS.

Like the automount daemon, the autofsd daemon creates symbolic links for file systems that are served locally. But if the autofsd daemon encounters an entry in a list of hierarchical file systems that is served locally and would result in a circular symbolic link on the local system (for example, a link from the /usr/local/bin directory back to itself), the group semantic is lost. AutoFS will mount and unmount the file systems on an individual basis.

This happens because AutoFS is designed to mount a remote file system on (or create a symbolic link to) the designated mount point itself. It does not, as Automount does, create an additional symbolic link back to a special temporary directory from which the remote file system is actually served. See Section A.4 for more information.

A.2.4 Shared Mounts

Shared mounts prevent duplicate mounts of the same remote file system when multiple subdirectories within it are accessed. Instead of creating a duplicate mount for each subdirectory, the daemon creates a symbolic link to the file system that is already mounted.

You can specify shared mounts by formatting the location field for each subdirectory entry as follows:

```
host:path:subdir
```

Note

AutoFS does not support this syntax. If the `autofsmount` command encounters this syntax, it converts the final colon (:) to a slash (/) and treats the entry as a typical AutoFS mount.

The `host` field is the remote host from which to mount the file system. The `path` field is the pathname of the directory to mount, and the `subdir` field, if specified, is the name of the subdirectory to which the symbolic link is made.

Suppose an indirect map called `/auto.myindirect` is specified in a master file as follows:

```
/mydir          /auto.myindirect
```

And the `/auto.myindirect` map consists of the following entries:

```
mybin           host1:/usr/staff/diane:bin
mystuff         host1:/usr/staff/diane:stuff
```

When a user accesses a file in `/mydir/mybin`, the `automount` daemon mounts `host1:/usr/staff/diane`, but creates a symbolic link called `/mydir/mybin` to the `bin` subdirectory in the temporarily mounted file system. If a user immediately tries to access a file in `/mydir/mystuff`, the `automount` daemon needs only to create a symbolic link that points to the `stuff` subdirectory because the `/usr/staff/diane` directory is already mounted. With the following map, the daemon would perform two separate mount operations:

```
mybin           host1:/usr/staff/diane/bin
mystuff         host1:/usr/staff/diane/stuff
```

A.2.5 Replicated File Systems

You can specify multiple locations for a single mount. If a file system is located on several servers and one of the servers is disabled, the file system can be mounted from one of the other servers. (This makes sense only when mounting a read-only file system, where there are no file changes to be synchronized.)

In the following example, the reference pages can be mounted from host1, machine2, or system3:

```
/usr/man\
                -ro,soft        host1:/usr/man \
                                machine2:/usr/man \
                                system3:/usr/man
```

The preceding example can also be expressed as a list of servers, separated by commas and followed by a colon and the pathname, for example:

```
/usr/man  –ro,soft  host1,machine2,system3:/usr/man
```

This syntax is valid only if the pathname is the same on each server.

When you access the reference pages, the `automount` daemon issues a ping request (NFS v2 loop request) to each of the specified servers concurrently and selects the first server that responds to serve the file system.

In contrast, the `autofsd` daemon first classifies each of the specified servers based on the proximity of its network address to the current system (Local, Same Subnet, Same Network, or Other Network). The daemon then attempts to serve the file system from the closest resource, starting with Local addresses.

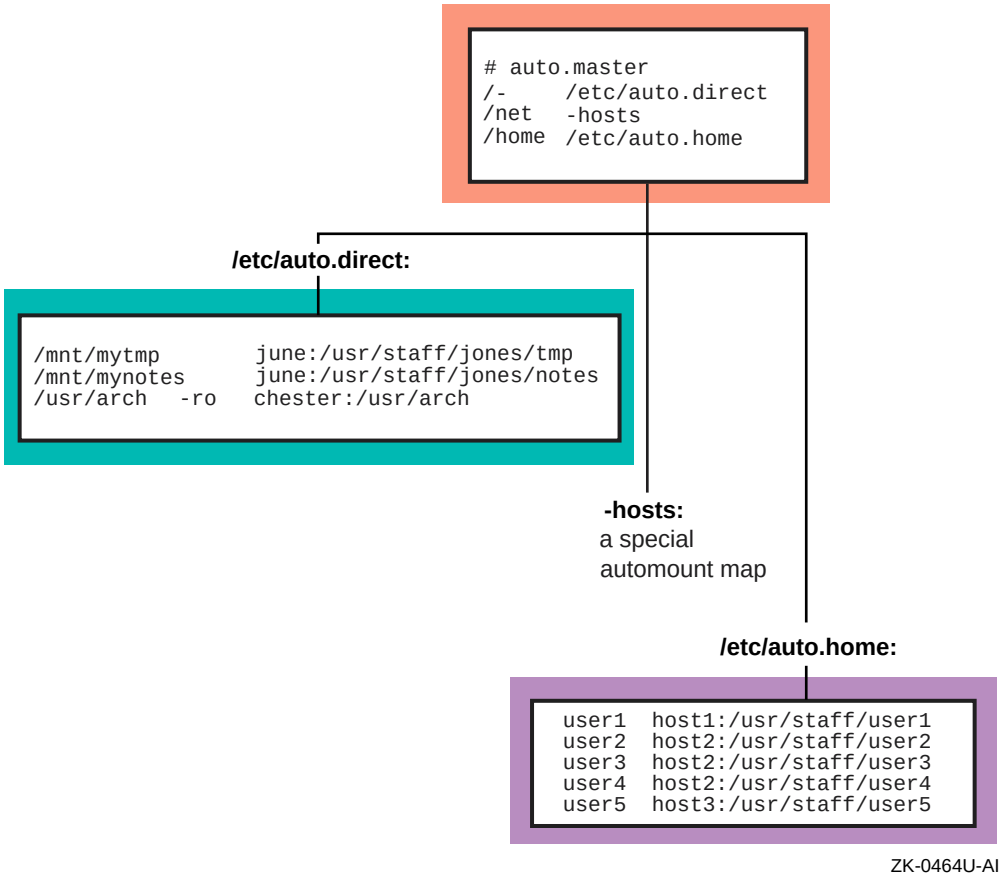
If the file system can be served locally, the daemon uses a symbolic link to access it. If the file system cannot be served locally, the daemon resorts to trying all Same Subnet, Same Network, and Other Network addresses, in that order. Except when checking Local addresses, the system issues a ping request to each server concurrently and selects the first server that responds to serve the file system.

A.3 Map Examples

The examples in this section illustrate how the same maps can be rewritten in a number of ways.

Figure A–1 illustrates an `auto.master` map that points to the `/etc/auto.direct` direct map, the built-in `–hosts` map, and the `/etc/auto.home` indirect map. Each map to which the `auto.master` map points is expanded to show its sample contents.

Figure A–1: Sample automount Maps



ZK-0464U-AI

The following examples show how the `/etc/auto.direct` map in Figure A–1 can be rewritten using multiple mounts (Example A–1); multiple mounts and shared mounts (Example A–2); and multiple mounts, shared mounts, and replicated file systems (Example A–3).

Example A–1: Multiple Mounts in a Direct Map

```
/mnt/mytmp                          june:/usr/staff/jones/tmp
/mnt/mynotes                        june:/usr/staff/jones/notes
/usr/arch       /           –ro     chester:/usr/arch \
                /bsd        –ro     chester:/usr/arch/bsd \
                /standards  –ro     chester:/usr/arch/standards \
                /dec/uws    –ro     chester:/usr/arch/dec/uws \
                /dec/ultrix –ro     chester:/usr/arch/dec/ultrix
```

Example A–2: Multiple Mounts and Shared Mounts in a Direct Map

```
/mnt/mytmp                          june:/usr/staff/jones:tmp
/mnt/mynotes                        june:/usr/staff/jones:notes
/usr/arch       /           –ro     chester:/usr/arch \
                /bsd        –ro     chester:/usr/arch/bsd \
                /standards  –ro     chester:/usr/arch/standards \
                /dec/uws    –ro     chester:/usr/arch/dec/uws \
                /dec/ultrix –ro     chester:/usr/arch/dec/ultrix
```

Example A–3: Multiple Mounts, Shared Mounts, and Replicated File Systems in a Direct Map

```
/mnt/mytmp                          june:/usr/staff/jones:tmp
/mnt/mynotes                        june:/usr/staff/jones:notes
/usr/arch       /           –ro     chester:/usr/arch \
                /bsd        –ro     chester:/usr/arch/bsd \
                                    bazel:/src/bsd \
                /standards  –ro     chester:/usr/arch/standards \
                /dec/uws    –ro     chester:/usr/arch/dec/uws \
                                    fiesta:/archive/uws\
                /dec/ultrix –ro     chester:/usr/arch/dec/ultrix
```

The `/etc/auto.direct` maps in the preceding examples could be rewritten as indirect maps. If the `/etc/auto.direct` map is rewritten to be an indirect map, the entry pointing to it in the `auto.master` map would look like the following:

```
/mnt     /etc/auto.indirect
```

Rewritten as a simple indirect map (`/etc/auto.indirect`), the `/etc/auto.direct` map in Figure A–1 would read as shown in Example A–4.

Example A–4: Simple Indirect Map

```
mytmp           june:/usr/staff/jones/tmp
mynotes         june:/usr/staff/jones/notes
arch    –ro     chester:/usr/arch
```

The following examples illustrate that indirect maps can also be rewritten using multiple mounts (Example A–5); multiple mounts and shared mounts (Example A–6); and multiple mounts, shared mounts, and replicated file systems (Example A–7).

Example A–5: Multiple Mounts in an Indirect Map

```
mytmp                                   june:/usr/staff/jones/tmp
mynotes                                 june:/usr/staff/jones/notes
arch            /               –ro     chester:/usr/arch \
                /bsd            –ro     chester:/usr/arch/bsd \
                /standards      –ro     chester:/usr/arch/standards \
                /dec/uws        –ro     chester:/usr/arch/dec/uws \
                /dec/ultrix     –ro     chester:/usr/arch/dec/ultrix
```

Example A–6: Multiple Mounts and Shared Mounts in an Indirect Map

```
mytmp                                   june:/usr/staff/jones:tmp
mynotes                                 june:/usr/staff/jones:notes
arch            /               –ro     chester:/usr/arch \
                /bsd            –ro     chester:/usr/arch/bsd \
                /standards      –ro     chester:/usr/arch/standards \
                /dec/uws        –ro     chester:/usr/arch/dec/uws \
                /dec/ultrix     –ro     chester:/usr/arch/dec/ultrix
```

Example A–7: Multiple Mounts, Shared Mounts, and Replicated File Systems in an Indirect Map

```
mytmp                                   june:/usr/staff/jones:tmp
mynotes                                 june:/usr/staff/jones:notes
arch            /               –ro     chester:/usr/arch \
                /bsd            –ro     chester:/usr/arch/bsd \
                                        bazel:/src/bsd \
                /standards      –ro     chester:/usr/arch/standards \
                /dec/uws        –ro     chester:/usr/arch/dec/uws \
                                        fiesta:/archive/uws\
                /dec/ultrix     –ro     chester:/usr/arch/dec/ultrix
```

As previously described, the –`hosts` map shown in Figure A–1 is a built-in map supplied by Automount and AutoFS. In this case, if a user switches into a directory called `/net/hosts1` on the system running Automount or AutoFS , the system checks for references to `host1` in the `hosts` database and attempts to mount all of the file systems exported by `host1` on the local system.

The `/etc/auto.home` map shown in Figure A–1 is an indirect map that allows users to remotely mount their home directories. If necessary, it can be rewritten using substitution characters, as follows:

```
user1  host1:/usr/staff/&
user2  host2:/usr/staff/&
```

```
user3  host2:/usr/staff/&
user4  host2:/usr/staff/&
user5  host3:/usr/staff/&
```

A.4 Understanding Automount and AutoFS Behavior

Automount and AutoFS are fundamentally the same in that both services allow you to automatically mount file systems on an as-needed basis. However, each service performs its function in a different manner that is mostly invisible to the end user.

The following sections describe:

- How each service mounts remote file systems
- How automatic mounts are triggered for each service

A.4.1 Mounting Remote File Systems

The primary difference between Automount and AutoFS is that Automount serves remote file systems by mounting them in a temporary directory on the local system (`/tmp_mount`, by default) and creating symbolic links from that directory to the intended mount point. In contrast, AutoFS mounts remote file systems directly on their intended mount points on the local system (unless the file system exists on the local system itself, in which case, AutoFS creates a symbolic link).

Consider the following direct map example:

```
/tools/bin   apollo:/usr/opt/bin
/tools/lib   apollo:/usr/opt/lib
/tools/man   apollo:/usr/opt/man
```

If this map is served by Automount, and a user switches into `/tools/bin` on the local system, Automount responds by mounting the remote file system, `/usr/opt/bin`, in the local `/tmp_mnt/apollo/tools/bin` directory, then it creates a symbolic link from that directory to the intended target, `/tools/bin`.

If the same map is served by AutoFS, and a user switches into `/tools/bin` on the local system, AutoFS responds by mounting the remote file system, `/usr/opt/bin`, directly on the intended local target directory, `/tools/bin`. (AutoFS creates a symbolic link, from `/usr/opt/bin` to `/tools/bin`, only if the local system is the apollo server itself.)

Each behavior has its benefits. For example, because Automount mounts remote file systems in a temporary directory, you can use it to take advantage of shared mounts to prevent duplicate mounts of the same remote file system. AutoFS must mount these remote file systems on an individual basis.

On the other hand, AutoFS is more efficient because it does not deal with the overhead of a temporary directory and symbolic links. It also provides higher availability than Automount. Although the `autofsd` daemon must be running for mounts or unmounts to be performed, if it is killed or becomes unavailable, exisiting auto-mounted NFS file systems continue to be available. If the `automount` daemon fails, all of its auto-mounted file systems become unavailable because the `automount` daemon is required for lookups.

A.4.2 Inducing Automatic Mounts

Because the underlying mechanics of Automount and AutoFS are different, invoking file processing commands on a directory that contains intercept points (the key objects that trigger auto-mounts) can have vastly different results for each service.

For example, because many file processing commands do not follow symbolic links by default, these commands will not follow the symbolic links from an intercept point to Automount's temporary directory. Therefore, these file processing commands are not likely to process the contents of file systems that are served by Automount, even if those file systems were auto-mounted prior to the commands' execution.

For AutoFS, if the file processing command is designed to induce an auto-mount by AutoFS, the command will always process the contents of the auto-mounted file system, because AutoFS mounts file systems directly on their intercept points. As previously mentioned, AutoFS creates a symbolic link only if the local host is serving the requested file system. This type of symbolic link is recognized by many file processing commands, and, for consistency, is treated the same as any other file system served by AutoFS.

The following examples describe Automount and AutoFS behavior for the same command invoked in each environment. Except where noted, when a command's behavior differs for the AutoFS environment, the intention is to conform to Open Network Computing (ONC+) standards for AutoFS.

In all cases, *`directory`* is a directory that contains direct intercept points (the key objects named in direct maps).

Note

Indirect intercept points do not induce auto-mounts unless specifically named via a command.

Invoking a command on a directory that is a direct or indirect intercept point itself always induces an auto-mount and the

subsequent processing of the auto-mounted file system's contents by the command that triggers the auto-mount.

- The `ls` command:

  `ls -al` *`/directory`*

  Automount induces auto-mounts, but AutoFS does not.

  `ls -R` *`/directory`*

  Automount does not trigger auto-mounts, and the `ls` command does not process any previously auto-mounted file systems.

  AutoFS triggers auto-mounts, and the `ls` command processes these auto-mounted directories.

- The `find` command:

  `find` *`/directory expression`*

  Automount does not induce auto-mounts, and the `find` command does not process any previously auto-mounted file systems.

  AutoFS induces auto-mounts and the `find` command searches these auto-mounted file systems.

- The `chown -R` command:

  `chown -R` *`/directory`*

  Both Automount and AutoFS induce auto-mounts. However, where the `chown` command changes the ownership of files and directories in the AutoFS file systems, it does not process the Automount file systems.

  The `chgrp` command produces similar results.

- The `chmod` command:

  `chmod -R` *`/directory`*

  Automount does not induce auto-mounts, and the `chmod` command does not process auto-mounted file systems.

  AutoFS induces auto-mounts and the `chmod` command changes the permissions of the files and directories in these auto-mounted file systems (with the exception of the the directory intercept point itself).

- The `du` command:

  `du /directory`

  Neither Automount nor AutoFS induce auto-mounts. The `du` command does not process any auto-mounted directories.

  In this case, AutoFS in inconsistent with Sun's implementation, which both induces auto-mounts and processes auto-mounted directories.

- The `vdump` command:

```
vdump -0u /directory
```

Neither Automount nor AutoFS induce auto-mounts. The vdump command does not process any previously auto-mounted file systems.

B

NIS ypservers Update Scripts

This appendix provides the following scripts for updating the ypservers map:

- addypserver — Adds a slave server (Section B.1)
- rmypserver — Removes a slave server (Section B.2)

B.1 Add Slave Server Script

Use the following procedure to create the addypserver script on an NIS master server:

1. Create an addypserver file in the /var/yp directory and insert the following lines:

```
#!/bin/sh
PATH="/usr/bin:/var/yp:$PATH"
if [ $# != 1 ]; then
        echo "usage: $0 server"; exit 1
fi
DOMAIN=‘/usr/sbin/rcmgr get NIS_DOMAIN‘
METHOD=‘/usr/sbin/rcmgr get NIS_SERVERARGS d‘
cd /var/yp
echo "
Adding $1 to ypservers map for domain DOMAIN ..."
(/var/yp/makedbm -a $METHOD -u $DOMAIN/ypservers;\
echo $1 $1) | /var/yp/makedbm -a $METHOD $DOMAIN/ypservers
/var/yp/yppush ypservers
```

2. Set the permissions to 700, using the chmod command as follows:

```
# chmod 700 /var/yp/addypserver
```

To add host1 to the ypservers map, enter the following command:

```
# /var/yp/addypserver host1
```

You still need to edit the NIS master server's hosts source file and add an entry for the slave server, if it is not already in the hosts file. Then, update and distribute the hosts map by entering the make command. See Section 3.4.1 for more information.

B.2 Remove Slave Server Script

Use the following procedure to create the `rmypserver` script on an NIS master server:

1. Create a `rmypserver` file in the `/var/yp` directory and insert the following lines:

```
#!/bin/sh
PATH="/usr/bin:/var/yp:$PATH"
if [ $# != 1 ]; then
        echo "usage: $0 server"; exit 1
fi
DOMAIN=‘/usr/sbin/rcmgr get NIS_DOMAIN‘
METHOD=‘/usr/sbin/rcmgr get NIS_SERVERARGS d‘
cd /var/yp
echo "
Removing $1 from ypservers map for domain DOMAIN ..."
/var/yp/makedbm -a $METHOD -u $DOMAIN/ypservers | grep -v "^$1 " \
| /var/yp/makedbm -a $METHOD $DOMAIN/ypservers
/var/yp/yppush ypservers
```

2. Set the permissions to 700, using the `chmod` command as follows:

```
# chmod 700 /var/yp/rmypserver
```

To remove host1 from the `ypservers` map, enter the following command:

```
# /var/yp/rmypserver host1
```

You still need to edit the NIS master server's `hosts` source file and remove the entry for the slave server. Then, update and distribute the `hosts` map by entering the `make` command. See Section 3.4.2 for more information.

C

NFS Error Messages

You might see the following types of NFS error messages:

- Server error messages (Section C.1)
- Client error messages (Section C.2)

C.1 Server Error Messages

The following error messages are issued to the screen or console or sent to the `syslogd` daemon.

`authget: unknown authflavor` *n*
`authflavor`

Explanation: Each NFS request has an authentication type. This message is displayed if the type is not AUTH_UNIX.

User Action: Have the client application use the AUTH_UNIX authentication type.

`fh3tovp: bad length:` *n*

Explanation: A client sent a bad file handle to the server.

`NFS request from unprivileged port, source IP address =` *n*

Explanation: The server, performing NFS server port monitoring, received an NFS request from a nonprivileged port (greater than or equal to 1024) on a client. This might indicate a security problem.

`NFS server: fs(`*n,n* `) not mounted; client address =` *n.n.n.n*

Explanation: The client requested a file on a file system that is not mounted or does not exist on the server. This can occur if a file system is unmounted while clients are using it or if the client passed an invalid file handle.

User Action: Make sure that the appropriate file system is mounted on the NFS server. If the file system is mounted on the same device, have the client system retry the operation. If the file system is mounted on a different device, have the client system unmount and remount the remote file system.

`NFS server:  stale file handle fs(`*n,n* `) file` *file* `gen` *n*, `client address =` *n.n.n.n* `errno` *n*

Explanation: The client accessed a file that no longer exists. The file was deleted either by the server or by another client.

`NFS server:  unexported fs(`*n,n* `) file` *file*, `client address =` *n.n.n.n*

Explanation: A client that previously had access to a file system can no longer access the file system, either because of changes in the `/etc/exports` file or in the net group mapping.

User Action: Have the client system unmount the file system.

`rfs_dispatch botch`

Explanation: The duplicate request cache routine returned an illegal value.

`rfs_dispatch:  bad rfs reply` *n* `ret`

Explanation: A server routine did not return a value or returned an incorrect value.

`rfs_dispatch:  dispatch error, no reply`
`rfs_dispatch:  sendreply failed`

Explanation: Possible reasons for this message include the following:

- The server is out of memory and cannot process or reply to a request.
- The server cannot find a route to the source.
- There is some other network-related problem.

`too many nfsds`

Explanation: There are more `nfsd` daemons registered with NFS than were started.

C.2 Client Error Messages

This appendix provides an explanation and suggested user actions for the following classes of client error messages:

- Remote mount error messages
- Automount error messages
- AutoFS error messages
- Console error messages

Within each section, error messages are listed alphabetically.

C.2.1 Remote Mount Error Messages

The following error messages are displayed if you are mounting directories or file systems from remote systems:

`mount: unknown special file or file system` *xxx*

Explanation: There is no entry in the `/etc/fstab` file for the mount point that you specified in the `mount` command line.

User Action: Verify that there is an entry in the `/etc/fstab` file for the file system. If not, add an entry. If one exists, look for syntax errors or typos in the entry. See `fstab`(4).

`/etc/fstab: No such file or directory`

Explanation: The `/etc/fstab` file does not exist. The `mount` command discovered this when it tried to look up the name specified on the command line.

User Action: Create an `/etc/fstab` file and include the appropriate entries. See `fstab`(4).

`nfs_mount: Permission denied for` *yyy*

Explanation: Your host name is not in the export list for the file system or directory you want to mount from the server.

User Action:

1. Get a list of your host's exported file systems and directories by using the `showmount -e` command. For example, enter the following command if your server's host name is host2:

   ```
   # /usr/bin/showmount -e host2
   ```

2. If the file system or directory you want to mount remotely is not on the list, or if your host or network group name is not on the user list for the file system or directory, log in to the server and look in the `/etc/exports` file for the correct file system entry.
3. If the file system or directory name is in the `/etc/exports` file, but not in the output from `showmount`, the failure is in the `mountd` daemon. The `mountd` daemon could not parse that line in the file, could not find the file system or directory, or the file system or directory name was not a locally mounted file system.

   If the file system or directory name is in the `/etc/exports` file and Network Information Service (NIS) is configured, verify

that the ypbind daemon is running; it might have stopped. See exports(4) for further information.

`nfs_mount: cannot mount` *xxx* `on` *yyy*`: Mount device busy`

Explanation: The file system or directory you are trying to mount is already mounted.

`nfs_mount: cannot mount` *xxx* `on` *yyy*`: No such file or directory`

Explanation: The local mount point does not exist.

User Action: Verify that the mount point exists and that it is spelled correctly.

`nfs_mount: cannot mount` *xxx* `on` *file*`: Not a directory`

Explanation: Either the remote file system or the local mount point is not a directory.

User Action: Verify that the remote file system and the local mount point are directories (not files) by using the `ls` command. Verify the spelling of both directories.

`nfs_mount: cannot mount` *xxx* `on` *yyy*`: Not owner`

Explanation: You must mount the remote file system or directory as superuser (root) on your system.

`nfs_mount: illegal file system name` *xxx*`; use host:pathname`

Explanation: You did not specify the name of the server when you issued the `mount` command.

User Action: For example, to mount the file system `/usr/src` from the server host2, enter the following command:

```
# mount host2:/usr/src /host2/usr/src
```

`nfs_mount: invalid directory name` *xxx*
`directory pathname must begin with '/'.`

Explanation: The mount point on the local (client) system must be an absolute path starting at the root directory (`/`).

`nfs_mount: RPC: Authentication error;`
`why=Client credential too weak`

Explanation: The server is allowing client superuser mounts only and you are not a superuser. See `mountd`(8) for further information.

```
nfs_mount:  RPC: Authentication error;
why=Server rejected credential
```

Explanation: Possible reasons for this error message include the following:

- The server is running with Internet address verification turned on and it cannot resolve your Internet address. If your system has multiple network interfaces configured, the server must be able to resolve all IP addresses, either using the local `/etc/hosts` file or the distributed `hosts` file.
- The server is running with domain or subdomain verification turned on and your system is not in the same domain or subdomain as the server.

See `mountd`(8) for further information.

```
nfs_mount:  xxx server not responding:  port mapper failure
rpc timed out Giving up on yyy
```

Explanation: The server you are trying to mount from is down, or its port mapper is inoperative.

User Action:

1. Log in remotely to the server. If you are able to log in, the network is working.
2. Execute the `rpcinfo` command from the server. For example, for a server named host2, you would enter the following command:

   ```
   # /usr/sbin/rpcinfo –p host2
   ```

   If the port mapper is running properly on the server, the `rpcinfo` command lists the registered program numbers. If it does not, restart the port mapper on the server. You also need a port mapper running on the client host; if it is not running there, start it. See `portmap`(8) for more information.
3. After you restart the port mapper, stop the NFS daemons by entering the following command:

   ```
   # /sbin/init.d/nfs stop
   ```

   If NIS is running, stop the `ypbind` daemon on the server. Use the `ps` command to obtain the process ID (PID) and the `kill` command to stop the process:

   ```
   # ps -A | grep ypbind
      439 ??       I        0:00.02 /usr/sbin/ypbind ...
   170866 pts/3    S  +     0:00.01 grep ypbind
   # kill -9 439
   ```

4. If you stopped the `ypbind` daemon, restart it by entering the following command:

   # **/usr/sbin/ypbind &**

   Restart the NFS daemons on the server by entering the following command:

   # **/sbin/init.d/nfs start**

`nfs_mount: xxx server not responding: rpc prog not registered`

Explanation: The `mount` command got through to the port mapper, but the NFS `mountd` daemon was not registered.

User Action:

1. Log in to the server.
2. Verify that the `/usr/sbin/mountd` file exists by using the `ls` command.
3. Run the `ps` command to see if the `mountd` daemon is running. If it is not running, restart it by entering the following command:

   # **/usr/sbin/mountd**

`Can't get net id for host`

Explanation: There is no entry in the `/etc/hosts` file for the NFS server specified in the `mount` command line. If NIS is running, there is no entry in the `hosts` NIS map for the host name specified. If BIND is running, there is no entry in the `hosts` database for the host name specified.

C.2.2 Automount Error Messages

The following error messages are issued to the screen or console or sent to the `syslogd` daemon by the `automount` program:

`bad entry in map` *`mapname`*

Explanation: The map entry in *`mapname`* is malformed and the `automount` program cannot interpret it.

User Action: Verify the entry; you might need to include escape characters.

`Can't mount` *`mountpoint`*`:` *`reason`*

Explanation: The `automount` program cannot mount itself at *`mountpoint`*. The error is indicated in the *`reason`* statement.

`couldn't create directory: reason`

Explanation: The `automount` program could not create a directory. The error is indicated in the *reason* statement.

`dir mountpoint must start with '/'`

Explanation: The *mountpoint* must have a full pathname.

User Action: Verify both the spelling and path name of the mount point.

`hierarchical mountpoint: mountpoint`

Explanation: The `automount` program will not allow itself to be mounted within an automounted directory.

User Action: Use another strategy to mount the directory.

`host hostname not responding`

Explanation: The `automount` program attempted to mount from *hostname* but received no response or failed. These errors could indicate a server or network problem.

hostname:filesystem `server not responding`

Explanation: The `automount` program attempted to mount from *hostname* but received no response or failed. These errors could indicate a server or network problem.

hostname`: exports: rpc_err`

Explanation: The `automount` program encountered an error while attempting to get the list of exported file systems and directories that it is allowed to mount from *hostname*.

This error occurs when a user attempts to access a mount point that has the –`hosts` map associated with it. This error indicates a server or network problem.

hostname:filesystem `already mounted on` *mountpoint*

Explanation: The `automount` program is attempting to mount a file system on a mount point that has already been mounted with that file system.

`map` *mapname*, `key` *key*: `bad`

Explanation: The map entry in *mapname* is malformed and the `automount` program cannot interpret it.

User Action: Verify the entry; you might need to include escape characters.

mapname: Not found

Explanation: The automount program cannot locate the map it requires. This message is returned only when you specify the –v option.

mapname: *yp_err*

Explanation: The automount program encountered an error when looking up an NIS map entry.

Mount of *hostname:filesystem* on *mountpoint:* reason

Explanation: The automount program attempted to mount from *hostname* but received no response or failed. These errors could indicate a server or network problem.

mountpoint: Not a directory

Explanation: The *mountpoint* exists but is not a directory.

User Action: Verify both the spelling and pathname of the mount point.

mountpoint– *pathname* from *hostname*: absolute symbolic link

Explanation: The automount program detected that *mountpoint* is an absolute symbolic link (begins with /). The content of the link is *pathname*. Because this might have undesired consequences on the client, the automount program will not mount on absolute symbolic links.

no mount maps specified

Explanation: The automount program cannot find any maps to serve, nor can it find any NIS maps. This message is returned only when you specify the –v option.

WARNING: *hostname:file system* already mounted on mountpoint

Explanation: The automount program is mounting itself on top of an existing mount point. This message is a warning only.

WARNING: *mountpoint* not empty!

Explanation: The *mountpoint* directory is not empty. This message is returned only when you specify the –v option. It is warning you that the previous contents of *mountpoint* will not be accessible while the mount is in effect.

The following error messages can occur when a file system is exported from multiple servers as specified in a multiple-server map entry. They indicate possible network problems that can occur when the automount daemon requests a response from the servers.

`Cannot create socket for broadcast rpc: rpc_err`

Explanation: No server in a multiple-server map entry is responding. This indicates that the replicated file system could not be reached on any of the specified servers.

`Cannot receive reply to many_cast: rpc_err`

Explanation: No server in a multiple-server map entry is responding. This indicates that the replicated file system could not be reached on any of the specified servers.

`Cannot send broadcast packet: rpc_err`

Explanation: No server in a multiple-server map entry is responding. This indicates that the replicated file system could not be reached on any of the specified servers.

`Many_cast select problem: rpc_err`

Explanation: No server in a multiple-server map entry is responding. This indicates that the replicated file system could not be reached on any of the specified servers.

`NFS server (pid n@mountpoint) not responding still trying`

Explanation: An NFS request to the automount daemon with PID *n* serving mount point has timed out. The automount daemon might be overloaded or not running.

User Action: If the condition persists, reboot the client. You can also do the following:

1. Exit all processes that are using automounted directories.
2. Kill the current automount process.
3. Restart the automount process from the command line.

`Remount hostname:filesystem on mountpoint server not responding`

Explanation: The automount program was attempting to remount *filesystem* because it discovered that a part of the automounted hierarchy at the *mountpoint* was busy. The remote file system's server, *hostname*, did not respond to the mount request. This error indicates a server problem.

`trymany: servers not responding: reason`

Explanation: No server in a multiple-server map entry is responding. This indicates that the replicated file system could not be reached on any of the specified servers.

C.2.3 AutoFS Error Messages

The following sections describe error messages for the two components of AutoFS: the `autofsd` daemon and the `autofsmount` command.

C.2.3.1 autofsd Messages

The following error messages are issued to the screen or console or sent to the `syslogd` daemon by the `autosfd` program:

`autofs not configured`

Explanation: AutoFS is not properly configured in the kernel.

User Action: If necessary, add the `AUTOFS` option to the kernel configuration file and rebuild the kernel. See *System Administration* for more information on modifying and rebuilding the kernel.

`Cannot create socket for nfs: reason`

Explanation: Network socket creation failed due to *reason*.

`can't mount hostname`

Explanation: A mount request was rejected by the `mountd` daemon on *hostname*. This error usually indicates a permissions problem or that the file system does not exist.

User Action: Verify the export permissions in the `/etc/exports` file on the server and verify that the file system exists.

`Can't ping mountd version NFS-version at server hostname reason`

Explanation: The `autofsd` daemon attempted to communicate with the `mountd` daemon on the *hostname* server, but received no response.

User Action: Verify the status of the network and verify that the server is properly configured and running NFS services.

`cfg_subsys_state returned errorcode`

Explanation: AutoFS is not properly configured in the kernel.

User Action: If necessary, add the `AUTOFS` option to the kernel configuration file and rebuild the kernel. See *System Administration* for more information on modifying and rebuilding the kernel.

`host` *`hostname`* `not responding`

Explanation: The `autofsd` daemon attempted to mount from *`hostname`* but it received no response or the request failed. These errors could indicate a server or network problem.

User Action: Verify the status of the network and verify that the server is properly configured and running NFS services.

`hostname` `exports:` *`rpc_err`*

Explanation: The `autofsd` daemon encountered an error while attempting to get the list of exported file systems and directories that is allowed to mount from *`hostname`*. This occurs during attempted access to a mount point with the `-hosts` map. It indicates a server or network problem.

User Action: Verify the status of the network and verify that the server is properly configured and running NFS services.

`hostname``:  mountd not responding` *`reason`*

Explanation: The `autofsd` daemon attempted to communicate with the `mountd` daemon on the *`hostname`* server, but received no response.

User Action: Verify the status of the network and verify that the server is properly configured and running NFS services.

`hostname``:  server's portmap not responding`

Explanation: The `autofsd` daemon attempted to communicate with the `portmap` daemon on the *`hostname`* server, but received no response.

User Action: Verify the status of the network and verify that the server is properly configured and running NFS services.

`lookup_addr:  gethostbyname failed` *`error`* `for` *`hostname`*

Explanation: The `autofsd` daemon was unable to obtain a network address for the named host.

User Action: Verify the address for the host in the local hosts file and the DNS or NIS database. Verify that the DNS or NIS server is up and running.

`match` *`mapname`*`:`*`keyname`* `failed:` *`reason`*

Explanation: The `autofsd` daemon is having a problem reading the map file *`mapname`* to find key *`keyname`*. The error is indicated in the *`reason`* statement.

`Mount of hostname:filesystem on mountpoint:  reason`

Explanation: The `autofsd` daemon attempted to mount from *hostname*, but it received no response or the request failed. These errors could indicate a server or network problem.

User Action: Verify the status of the network and verify that the server is properly configured and running NFS services.

`Unable to locally serve filesystem`

Explanation: Locally serving the file system would result in a circular symbolic link.

User Action: Choose a different mount point for the file system, or specify a different host to serve the file system.

C.2.3.2 autofsmount Messages

`autofs is not configured or not enabled`

Explanation: AutoFS is not properly configured in the kernel.

User Action: If necessary, add the `AUTOFS` option to the kernel configuration file and rebuild the kernel. See *System Administration* for more information on modifying and rebuilding the kernel.

`Intercept filesystem mount failed:  reason`

Explanation: The attempt to create an intercept mount point for *filesystem* has failed due to *reason*. The `autofsmount` command issues this error message for direct map entries, and when running in verbose mode, for indirect map entries as well.

`Map mapname does not exist`

Explanation: The `autofsmount` command could not find the specified direct or indirect map file.

User Action: Ensure that you have specified the proper location for the map files on the command line or in your master map file.

`Note:  Indirect entry in map mapname with key keyname cannot be locally served with the mounton and mountfrom directories as defined.`

Explanation: An external server will be chosen to avoid a circular symbolic link.

Note: The hierarchical entry in map *mapname* for *keyname* cannot be served locally

Explanation: An external server will be chosen to avoid a circular symbolic link.

Note: The shared map entry in map *mapname* with key *keyname* will be converted to a non-shared entry

Explanation: AutoFS does not support the shared mount syntax of Automount. It converts all shared map entries to their nonshared counterparts.

Unmount *filesystem*: *reason*

Explanation: An attempt to unmount *filesystem* has failed with *reason*.

Warning: Cannot support the hierarchy in map *mapname* with key *keyname* with the mounton and mountfrom directories as defined.

Explanation: The hierarchical direct map entry for subdirectory / cannot be supported, as no external servers are listed and locally serving it would create a circular symbolic link. The file systems in the map entry will be treated as though they are individual map entries.

User Action: Specify an external server or change the key and/or the location of the file system in question to avoid a circular symoblic link.

Warning: Skipping entry in map *mapname* with key *keyname*

Explanation: The file system will be locally served. No intercept mount point will be created, only a symbolic link.

Warning: The hierarchical entry in map *mapname* for *keyname* will not work.

Explanation: A hierarchical direct map entry for some subdirectory other than / cannot be supported, as no external servers are listed and locally serving it would create a circular symbolic link. The file systems in the map entry will be treated as though they are individual map entries.

User Action: Specify an external server or change the key and/or the location of the file system in question to avoid a circular symoblic link.

Warning: There are no servers available for this entry

Explanation: In the context of previous messages for this map entry, this error message indicates that locally serving the file system would create a circular symbolic link, and no external servers are specified.

User Action: Specify an external server or change the key and/or the location of the file system in question to avoid a circular symoblic link.

C.2.4 Console Error Messages

The following error messages might be displayed on the NFS client system console and in the error logger. They note an NFS file access failure.

`NFS server` *`hostname`* `not responding, still trying`

Explanation: File operations in a hard-mounted file system are suspended because communication between the client and the server has stopped.

`NFS server` *`hostname`* `ok`

Explanation: File operations have resumed.

`NFS` *`file operation`* `failed for server` *`hostname`*: *`reason`*

Explanation: If the operation is in a soft-mounted file system and the server is inoperable, the reason for the failure is that the operation timed out.

`NFS write error, server` *`hostname`*, `remote file system full`

Explanation: A write operation failed because the remote file system is full.

`NFS write error` *`errno`*, `server` *`hostname`*, `fs(`*`n,n`*`), file` *`file`*

Explanation: A write operation was refused by the server. The *`fs`* and *`file`* variables are parts of the file handle (fhandle). See `errno`(2) for a description of write errors.

D

uucp Messages

This appendix provides a description and suggested user actions for the following uucp messages:

- Status and log file messages (Section D.1)
- tip error messages (Section D.2)

D.1 Status and Log File Messages

The messages in this section might appear in uucp status or log files. Use the uulog or uustat command to see the status messages.

ASSERT ERROR

An ASSERT error occurred, indicating a condition that only a system manager can solve. ASSERT errors are stored in the /usr/spool/uucp/.Admin/errors file and have the following form:

ASSERT ERROR (*prog*) pid: *xxxx* (*date/time*) *error error-location*

The variables have the following meaning:

prog	Name of the program generating the error.
xxxx	Process ID (PID) of the program.
date/time	Data and time when the error occurred.
error	A message describing the error. The message might include arguments. If there is a value contained in parentheses following the message, this value is often the error number (errno).
error-location	Name and version of the source file and the line in the file where the error occurred.

Table D–1 lists the ASSERT error messages.

Table D–1: ASSERT Error Messages

Error Message	Explanation and User Action
BAD LINE *line* (*num*)	The /usr/lib/uucp/Devices file has a bad line: *line* is the bad line and *num* is the number of fields found in the line.
	Correct the entry in the file. See Devices(4) for information on the file entries.
BAD LOGIN_UID (–1) BAD UID (–1) CAN NOT FIND UID (*num*)	The user ID used by the process is not currently logged in and is not defined in the /etc/passwd file or the networks database, if using NIS.
	Check your user ID by using the id command, and change the entry in the /etc/passwd file or the networks database, if using NIS.
BAD SPEED (*num*)	An unsupported baud rate (*num*) was specified.
	Check the command arguments or uucp configuration files. Then run uucpsetup to change the baud rate.
CAN'T CHDIR *dir* (*num*)	A command to change to directory *dir* failed with errno *num*. The uucp program required read access to the directory.
	Check the permissions on the directory. If the directory does not exist, check the permissions on the spool directory.
CAN'T CLOSE file (*num*)	Could not close file with errno *num*.
CAN'T CREATE file (*num*)	Could not open file with errno *num*. The uucp program needs write access to the file or directory.
	Check the permissions on the file and directory.
CAN'T LINK file (*num*)	Could not link a source file to the work file *file* in the uucp spool directory with errno *num*.
	Check the spool directory permissions.
CAN'T LOCK LCK.SQ. *sys* (0)	Could not lock the /var/spool/locks/LCK.SQ. *sys* file for system *sys*.
	Check the time and permissions on the file. If it is old, delete the file.

Table D–1: ASSERT Error Messages (cont.)

Error Message	Explanation and User Action
CAN'T OPEN file (*num*)	Could not open file with errno *num*. The uucp program needs write access to the file or directory.
	Check the permissions on the file and directory.
CAN'T STAT file (*num*)	The uucico daemon could not obtain information about the file with errno *num*.
	Check the permissions on the file.
CAN'T UNLINK file (*num*)	Could not unlink the file with errno *num*.
	Check the permissions on the file.
CAN'T WRITE file (*num*)	Could not open the file with errno *num*. The uucp program needs write access to the file or directory.
	Check the permissions on the file and directory.
FILE EXISTS file (*num*)	The file already exists and an access() call on that file returned errno *num*. The file is a uucp work file that was not cleaned up by another uucp process.
No uucp server (0)	The uucp service is not defined in the /etc/services file.
	Edit the /etc/services file and add a uucp entry.
SYSLST OVERFLOW (*num*)	There are too many jobs queued for a single system. The number of jobs is *num*.
	Use the uustat –q command and examine the queue. If the jobs are not old, try the request again. If there are old jobs in the queue, use the uucleanup command to clean out the queue. See uucleanup(8) for more information.
TOO MANY LOCKS (*num*)	The system limit on the number of lock files was exceeded while creating lock file *num*.
	Retry the request after the the current activity is completed.

Table D–1: ASSERT Error Messages (cont.)

Error Message	Explanation and User Action
XMV ERROR file (*num*)	The uuxqt daemon could not move the execute file to the .Xqtdir directory in the uucp spool area and failed with errno *num*. Use the ls –l command and verify that the .Xqtdir directory is owned by uucp and has a 775 permission.

BAD LOGIN/MACHINE COMBINATION

Explanation: There are two possible reasons for this message:

- The VALIDATE option for the local system is set in the Permissions file on the remote system and the local system's user name does not match the LOGNAME entry for the system in the remote system's Permissions file.
- The local system's user name has no corresponding LOGNAME entry in the remote system's Permissions file.

User Action: Either ask the remote system administrator to add a LOGNAME entry for that user name, or edit the Systems file and modify the entry for the remote system to use a known user name.

BAD SEQUENCE CHECK

Explanation: The information in /usr/lib/uucp/SQFILE file on the local and remote system is inconsistent. Possible reasons include:

- A new SQFILE has been installed on either system, possibly because a new operating system release was installed.

  User Action: Synchronize the local and remote files.

- Another system is imitating either the local or remote system. This indicates a potential security problem.

  User Action: Verify that both systems are legitimate and report security issues, as necessary.

CALLBACK REQUIRED

Explanation: The local system initiated a call and informed the remote system that it has work for that system. The remote system is configured to accept work only if it initiates a call to the local system. Work is queued until the remote system calls the local system.

User Action: Monitor the queue to verify that all jobs are completed.

CALLER SCRIPT FAILED

Explanation: An error occurred while processing the chat script, defined in the `Systems` file.

User Action: Enter the `uutry` *remote_system* command and observe the prompts from the remote system. Compare the prompts to the chat script. If there is a difference, run the `uucpsetup` script and change the chat script.

CAN'T ACCESS DEVICE

Explanation: Possible reasons include:

- The physical device could not be opened.

  User Action: Check the permissions on the terminal (tty) line, using the `ls -l` command. If neither user `uucp` nor group `uucp` has write access to the line, change the mode to 666.

- The modem type is not defined in the `/usr/lib/uucp/Dialers` file.

  User Action: Verify that the modem type has an entry in the `Dialers` file. If not, run the `uucpsetup` script and make an entry for the modem type.

CANNOT OPEN SYSTEMS FILE FOR READ

Explanation: The `uucp` program cannot read the `/usr/lib/uucp/systems` file.

User Action: Change the mode to 650, and the owner and group to `uucp`.

CONN FAILED (*string*)

Explanation: The connection to the remote system failed; *string* describes the reason for the failure. The system will reconnect as necessary.

User Action: Monitor the queue to verify that all jobs are completed. If the problem persists, check your configuration.

CONVERSATION FAILED

Explanation: The conversation with the remote system has abnormally ended. Possible reasons are a modem error or system crash. Partially completed jobs are requeued and processed later.

User Action: Monitor the queue to verify that all jobs are completed.

DEVICE LOCKED

Explanation: Another utility (`tip`, `cu`, `uugetty`, or `uucico`) is already using the device.

User Action: Retry the request; you will continue to receive this message until the other utility has finished using the device.

DIAL FAILED

Explanation: The modem dialing sequence failed or timed out.

User Action: Retry the command.

LOGIN FAILED

Explanation: The `uucico` daemon timed out while trying to log in to the remote system.

User Action: Use the `uutry` command with your request to determine why the login is failing.

If the error occurs while processing the chat script, run the `uucpsetup` script and modify the chat script to reflect the actual messages used by the remote system. For example, if the chat script stops while waiting for a login prompt, modify the chat script to send a carriage return and delay before getting a login prompt.

If the login to the remote system is successful and then an error occurs, the `uucico` daemon on the remote system failed to start or was slow in sending the `Shere` message to the local system.

LOST LINE (LOGIN)

Explanation: The connection was lost during the login process.

User Action: Retry the request.

NO DEVICES AVAILABLE

Explanation: There are no devices available on this system of the type or speed requested.

User Action: You can install additional devices on your system, if your system allows, or modify the request to use one of the available devices in the `/usr/lib/uucp/Devices` file.

REMOTE DOES NOT KNOW ME

Explanation: The local system does not have an entry in the remote system's `Systems` file.

User Action: Contact the remote system's administrator to have an entry for your system put in the `Systems` file.

REMOTE HAS A LCK FILE FOR ME

Explanation: The remote system is trying to contact the local system while the local system is trying to connect to the remote system. The uucp utilities do not allow simultaneous connections between systems.

User Action: You can either retry the request later, or wait and see if the queued request is performed when the remote system connects to your system.

REMOTE REJECT AFTER LOGIN

Explanation: After successfully logging in to the remote system, the local and remote systems could not start a conversation. The remote system also returns the message BAD LOGIN/MACHINE COMBINATION.

User Action: Check the configuration for the connection on both systems.

REMOTE REJECT, UNKNOWN MESSAGE

Explanation: The remote system rejected the connection to the local system, but did not return a recognizable error message.

User Action: Retry your operation.

STARTUP FAILED

Explanation: After successfully logging in to the remote system, the local and remote systems could not start a conversation. Either the systems could not agree on a protocol or they could not start the protocol.

User Action: Verify that both the local and remote systems specify the same protocol in the /usr/lib/uucp/Systems file.

SUCCESSFUL

Explanation: The conversation completed successfully.

SYSTEM NOT IN Systems FILE

Explanation: The remote system is not in the /usr/lib/uucp/Systems file.

User Action: Use the uuname command to view a list of known uucp systems.

TALKING

Explanation: The local system is having a conversation with the remote system.

WRONG MACHINE NAME

Explanation: The remote system name does not match the system name entry in the /usr/lib/uucp/Systems file.

User Action: Verify the system name and run uucpsetup to make the necessary changes.

WRONG TIME TO CALL

Explanation: The remote system cannot be called at this time. The job is queued for completion later.

User Action: If you want to change the time, run uucpsetup.

D.2 tip Error Messages

The following messages might be displayed when using the tip utility:

all ports busy

Explanation: All ports are in use.

User Action: Try your request again later.

can't open log file '/var/log/aculog' for update
contact your administrator

Explanation: The /var/log/aculog file does not exist.

User Action: Create the file with the mode 664, and owner and group uucp.

/etc/phones: can't open phone numbers file

Explanation: The /etc/phones file does not exist, or the tip utility cannot read the phones file.

User Action: Verify that the phones file exists and that it is not corrupted. If necessary, create a new phones file. See phones(4) for more information.

link down

Explanation: The terminal line (tty) cannot be opened.

User Action: Check that the mode of the tty device is 666.

missing phone number

Explanation: The remote system's phone number is not in the /etc/phones file.

User Action: Edit the /etc/phones file and add the remote system's phone number.

`system_name:  missing device spec`

Explanation: The terminal line (dv parameter) is not defined in the /etc/remote file.

User Action: Edit the /etc/remote file and add the parameter.

`tip:  unknown host` *sysname*

Explanation: The remote host system is not in the /etc/remote file.

User Action: Do one of the following:

- Create an entry for the system in the /etc/remote file. See remote(4) for more information.
- Invoke tip using the remote host system's phone number instead of its name.

`tip:  can't open host description file`

Explanation: The /etc/remote file does not exist, or the tip utility cannot read the remote file.

User Action: Verify that the remote file exists and that it is not corrupted. If necessary, create a new remote file. See remote(4) for more information.

`tip:  unknown host tip`*speed*

Explanation: The tip utility is not configured to use the *speed* specified on the command line.

User Action: Verify whether the hardware supports the speed. If it can, create a tipspeed entry for the speed in the /etc/remote file, using other tipspeed entries as a model. Create corresponding UNIX-speed and dialspeed entries in the file. Specify the modem type and the serial port to which it is attached, using the at and dv fields in the dialspeed entry.

`Unknown ACU type`

Explanation: The modem is unsupported.

User Action: Check the at field for the host system entry in the /etc/remote file. If the entry is correct, create an entry for the modem in the /etc/acucap file. See acucap(4) for more information.

xxx: unknown parity value

Explanation: The parity value (pa parameter) in the /etc/remote file is invalid.

User Action: Edit the /etc/remote file and enter a valid value. See remote(4) for more information.

E

sendmail Error Messages

This appendix provides an explanation and suggested user actions for the `sendmail` error messages. These messages can occur when sending mail to another user on the same host or when sending mail using TCP/IP. If other mailers are configured on your system (for example, DECnet), see the documentation that accompanies the mailer for additional messages.

The following `sendmail` messages are returned in a rejected mail message or sent to the `syslogd` daemon:

`binmail: opening /usr/spool/mail/`*`filename`*` -: Permission denied`

Explanation: The `/bin/mail` program could not deliver the mail on the destination host.

User Action:

- Verify the permissions on the `/usr/spool/mail` directory. The correct permissions are 1777.
- Verify the mailbox permissions. The correct permissions are 600.
- Verify that the mailbox owner is correctly specified.

`Cannot send message for 3 days`

Explanation: The message was not delivered during the period specified by the retry parameter in the `/var/adm/sendmail/sendmail.cf` file. It is being returned to the sender. Possible reasons are as follows:

- The destination host does not exist.
- The mail was addressed to a host outside of your company and no relay host has been configured in the `/var/adm/sendmail/sendmail.cf` file.
- The host has been off line or the network connection has been unreliable for three days.

User Action:

1. Verify all address information.

2. If the mail was addressed to a host outside of your company, you might not be able to send the mail directly. Check your `sendmail` configuration by entering the following command:

   # **grep '^define(_GateINET' /var/adm/sendmail/*hostname*.m4**

   If the braces in the output are empty (that is, do not contain a host name), reconfigure `sendmail` and specify a relay host. See Section 7.3 for more information on specifying a relay name.

3. Send the message again. The message is queued and sent automatically when the host is reachable.

`Connection refused`

Explanation: The `sendmail` daemon is not running on the destination host.

User Action: Check whether `sendmail` is running on the host by using the `ps` command as follows:

ps –ax | grep send

If it is not, ask the system administrator to start `sendmail`.

`Connection timed out during user open`

Explanation: A problem occurred during the Simple Mail Transfer Protocol (SMTP) session between 2 hosts, causing a time out.

User Action: No user action is necessary; the message will be retried later.

`Host unknown`

Explanation: Possible reasons are as follows:

- An address record for the host was not found.
- The `/var/adm/sendmail/sendmail.cf` file does not define a relay host that can handle mail addresses outside of your company.

User Action:

1. If the Domain Name System (DNS) is not configured on your host, verify that the host's address is defined. Check the `/etc/hosts` file if you are resolving addresses locally or issue the `ypmatch` *`hostname`* `hosts` command if you are using the Network Information Service (NIS). The hosts entry in the `svc.conf` file defines the services used. If the host is not defined, ask your system administrator to correct the problem.
2. Check for MX records for the host by using the `nslookup` command as follows:

```
# nslookup –q=mx hostname
```

If a record exists, go to step 3.

3. Check for address records by using the `nslookup` command. If the address is not found, have the DNS administrator for the destination domain add an address record for the host in the destination domain's DNS data files.
4. If the mail was addressed to a host outside of your company, you might not be able to send the mail directly. Check your `sendmail` configuration by entering the following command:

```
# grep '^define(_GateINET' /var/adm/sendmail/hostname.m4
```

If the braces in the output are empty (that is, do not contain a host name), reconfigure `sendmail` and specify a relay host. Send the message again. See Section 7.3 for more information on specifying a relay name.

`I refuse to talk to myself`

Explanation: The local host was asked to connect to itself and deliver a message.

User Action: Check your `sendmail` configuration by entering the following command:

```
# grep '^define(_GateINET' /var/adm/sendmail/hostname.m4
```

If the braces on any line in the output contain your host's name, there is a configuration error. Reconfigure `sendmail`. See Section 7.3 for more information.

`Remote protocol error`

Explanation: This message is generally found in the `mail.log` file generated by the `syslogd` daemon and indicates a problem in communicating with the remote host.

User Action: No user action is necessary; the message will be retried later.

`Service unavailable`

This is a secondary error message. Some other error has occurred that caused `sendmail` to interpret an address as an action.

User Action: Look for other error messages, for example `Host unknown`, and resolve them first. Resolving other errors should resolve this error as well.

`User unknown/Addressee unknown`

Explanation: The message reached the final destination, but the user address was not found in the local `aliases` file or the local password file at the final destination.

User Action: Verify that the user's address is correct.

F

Host Resources MIB Implementation

The Tru64 UNIX Simple Network Management Protocol (SNMP) agent implements the Host Resources MIB as described in RFC 1514. Although the RFC describes conceptual objects for management of host systems, it describes them in very general terms.

This appendix describes the Tru64 UNIX Host MIB implementation, including each group or table defined in RFC 1514 (with sample data). The formatting of SNMP data is specific to the implementation of an application. HP currently does not ship an application that presents SNMP data in this manner with Tru64 UNIX.

F.1 Tru64 UNIX Implementation Summary

The basic Tru64 UNIX implementation of RFC 1514 is as follows:

- The RFC specifies that when a product registry does not exist, all MIB variables of type `ProductID` return an object identifier of 0.0.
- The values of the `hrDeviceIndex` and `hrFSIndex` parameters remain unique between system reboots.
- Write access is not implemented for any Host MIB object.

F.2 System Group

The system group object implementation notes are as follows:

- The `hrSystemInitialLoadDevice` parameter is not implemented.
- The `hrSystemInitialLoadParameters` parameter returns the name of the booted kernel.

The following are sample data:

```
{hrSystemUptime.0               , TimeTicks, 0d 23:00:20.00}
{hrSystemDate.0                 , OCTET STRING, 1995-11-28,15:31:52.01}
{hrSystemInitialLoadParameters.0 , OCTET STRING, vmunix}
{hrSystemNumUsers.0             , Gauge, 0}
{hrSystemProcesses.0            , Gauge, 20}
{hrSystemMaxProcesses.0         , INTEGER, 1024}
```

F.3 Storage Group

The operating system software represents three types of logical storage: swap space, kernel memory, and file systems. The storage group object implementation is as follows:

- One entry in the hrStorageTable group is the total kernel memory being used.
- One entry is the current total swap space. (The value of the hrStorageAllocationFailures parameter for this entry is always 0.)
- There are several entries that each describe a specific type of kernel memory (the kernel malloc table). There is an entry for each memory type returned by TBL_MALLOCTYPES on that particular host.

Note

These entries do not represent actual fixed-size memory pools that could be exhausted. They do, however, indicate how system memory is being utilized among the various subsystems.

The value of the hrStorageSize parameter for the kernel memory entries is always 0, because there is no actual limit.

- There is one entry in the hrStorageTable group for each locally mounted file system. As specified in RFC 1514, remotely mounted file systems are not represented in the hrStorageTable group.
- The value of the hrStorageDescr parameter for file system-related entries is the same as the the hrFSMountedPoint parameter for the same file system in the hrFSTable group.
- The values of the hrStorageIndex parameter for file system-related entries is returned in the hrFSStorageIndex variable for the same file system in the hrFSTable group.
- The value of the hrStorageType parameter for file system storage entries is always hrStorageOther.

See Section F.5 for information on the file system implementation.

The following are sample storage group data:

```
{hrStorageIndex.1               , INTEGER, 1}
{hrStorageType.1                , OBJECT IDENTIFIER, hrStorageRam}
{hrStorageDescr.1               , OCTET STRING, Total Kernel Memory}
{hrStorageAllocationUnits.1     , INTEGER, 1024}
{hrStorageSize.1                , INTEGER, 2088960}
{hrStorageUsed.1                , INTEGER, 261112}
{hrStorageAllocationFailures.1  , Counter, 0}
{hrStorageIndex.2               , INTEGER, 2}
```

```
{hrStorageType.2                , OBJECT IDENTIFIER, hrStorageVirtualMemory}
{hrStorageDescr.2               , OCTET STRING, Total Swap Space}
{hrStorageAllocationUnits.2     , INTEGER, 1024}
{hrStorageSize.2                , INTEGER, 200704}
{hrStorageUsed.2                , INTEGER, 11920}
{hrStorageAllocationFailures.2  , Counter, 0}
{hrStorageIndex.3               , INTEGER, 3}
{hrStorageType.3                , OBJECT IDENTIFIER, hrStorageRam}
{hrStorageDescr.3               , OCTET STRING, MBUF}
{hrStorageAllocationUnits.3     , INTEGER, 1}
{hrStorageSize.3                , INTEGER, 0}
{hrStorageUsed.3                , INTEGER, 4096}
{hrStorageAllocationFailures.3  , Counter, 0}
{hrStorageIndex.4               , INTEGER, 4}
{hrStorageType.4                , OBJECT IDENTIFIER, hrStorageRam}
{hrStorageDescr.4               , OCTET STRING, MCLUSTER}
{hrStorageAllocationUnits.4     , INTEGER, 1}
{hrStorageSize.4                , INTEGER, 0}
{hrStorageUsed.4                , INTEGER, 32768}
{hrStorageAllocationFailures.4  , Counter, 0}
   .
   .
{hrStorageIndex.99              , INTEGER, 99}
{hrStorageType.99               , OBJECT IDENTIFIER, hrStorageOther}
{hrStorageDescr.99              , OCTET STRING, /}
{hrStorageAllocationUnits.99    , INTEGER, 1024}
{hrStorageSize.99               , INTEGER, 63167}
{hrStorageUsed.99               , INTEGER, 46098}
{hrStorageAllocationFailures.99 , Counter, 0}
{hrStorageIndex.100             , INTEGER, 100}
{hrStorageType.100              , OBJECT IDENTIFIER, hrStorageOther}
{hrStorageDescr.100             , OCTET STRING, /proc}
{hrStorageAllocationUnits.100   , INTEGER, 8192}
{hrStorageSize.100              , INTEGER, 0}
{hrStorageUsed.100              , INTEGER, 0}
{hrStorageAllocationFailures.100, Counter, 0}
{hrStorageIndex.101             , INTEGER, 101}
{hrStorageType.101              , OBJECT IDENTIFIER, hrStorageOther}
{hrStorageDescr.101             , OCTET STRING, /usr}
{hrStorageAllocationUnits.101   , INTEGER, 1024}
{hrStorageSize.101              , INTEGER, 866102}
{hrStorageUsed.101              , INTEGER, 596323}
{hrStorageAllocationFailures.101, Counter, 0}
```

F.4 Device Tables

This implementation supports CPUs, network interfaces, and disks in the device-related tables; printers are not supported. The CPU support is as follows:

- Each CPU physically attached to the system is represented in both the `hrDevice` and `hrProcessor` tables.
- The value of the `hrDeviceErrors` parameter is always 0.
- The value of the `hrDeviceStatus` parameter is either `running` or `down`.
- The value of the `hrProcessorLoad` parameter is accurately determined for each processor running on the system. Processor idle time is any

time spent in the IDLE or WAIT states. Busy time is time spent in any other state.

A background task records CPU time every 30 seconds, retaining 2 snapshots. When an SNMP request is received, CPU times are fetched immediately and the load average is calculated as the difference between this current data and the least recent snapshot. In this manner the values returned for the hrProcessorLoad parameter are current load averages over a period of at least 30 seconds, but not more than 1 minute. The value of the hrProcessorLoad parameter is calculated as follows:

(delta *busy* / (delta *busy* + delta *idle*)) * 100

The disk support is as follows:

- Each disk is represented in the hrDeviceTable group, the hrdiskStorageTable group, and the hrPartitionTable group.
- The value of the hrDeviceStatus parameter is running if the disk is online, or down if the disk is offline.
- The value of the hrDeviceErrors parameter is the sum of hard and soft errors reported for the disk.
- The value of the hrPartitionFSIndex parameter is either zero (0) or the value of the hrFSIndex parameter for the hrFSTable entry corresponding to the offline file system.

The network device support is as follows:

- Each network interface is represented in both the hrDeviceTable group and hrNetworkTable group.
- The value of the hrDeviceStatus parameter is running if the interface is running, down if the interface is not up, or unknown.
- The value of the hrDeviceErrors parameter is the sum of inbound and outbound packet errors on that interface.
- The value of the hrNetworkIfIndex parameter is the same as the MIB-II value of the ifIndex parameter for that interface.

The following are sample device table data:

```
{hrDeviceIndex.1               , INTEGER, 1}
{hrDeviceType.1                , OBJECT IDENTIFIER, hrDeviceProcessor}
{hrDeviceDescr.1               , OCTET STRING, Digital 2100 Server Model A500MP}
{hrDeviceID.1                  , OBJECT IDENTIFIER, 0.0}
{hrDeviceStatus.1              , INTEGER, running}
{hrDeviceErrors.1              , Counter, 0}
{hrDeviceIndex.2               , INTEGER, 2}
{hrDeviceType.2                , OBJECT IDENTIFIER, hrDeviceProcessor}
{hrDeviceDescr.2               , OCTET STRING, Digital 2100 Server Model A500MP}
{hrDeviceID.2                  , OBJECT IDENTIFIER, 0.0}
{hrDeviceStatus.2              , INTEGER, running}
{hrDeviceErrors.2              , Counter, 0}
{hrDeviceIndex.3               , INTEGER, 3}
```

```
{hrDeviceType.3               , OBJECT IDENTIFIER, hrDeviceProcessor}
{hrDeviceDescr.3              , OCTET STRING, Digital 2100 Server Model A500MP}
{hrDeviceID.3                 , OBJECT IDENTIFIER, 0.0}
{hrDeviceStatus.3             , INTEGER, running}
{hrDeviceErrors.3             , Counter, 0}
{hrDeviceIndex.4              , INTEGER, 4}
{hrDeviceType.4               , OBJECT IDENTIFIER, hrDeviceProcessor}
{hrDeviceDescr.4              , OCTET STRING, Digital 2100 Server Model A500MP}
{hrDeviceID.4                 , OBJECT IDENTIFIER, 0.0}
{hrDeviceStatus.4             , INTEGER, running}
{hrDeviceErrors.4             , Counter, 0}
{hrDeviceIndex.5              , INTEGER, 5}
{hrDeviceType.5               , OBJECT IDENTIFIER, hrDeviceNetwork}
{hrDeviceDescr.5              , OCTET STRING, tu0 - DEC TULIP Ethernet Interface}
{hrDeviceID.5                 , OBJECT IDENTIFIER, 0.0}
{hrDeviceStatus.5             , INTEGER, running}
{hrDeviceErrors.5             , Counter, 9}
{hrDeviceIndex.6              , INTEGER, 6}
{hrDeviceType.6               , OBJECT IDENTIFIER, hrDeviceNetwork}
{hrDeviceDescr.6              , OCTET STRING, tra0 - DEC DW300 Token Ring Interface}
{hrDeviceID.6                 , OBJECT IDENTIFIER, 0.0}
{hrDeviceStatus.6             , INTEGER, down}
{hrDeviceErrors.6             , Counter, 0}
{hrDeviceIndex.7              , INTEGER, 7}
{hrDeviceType.7               , OBJECT IDENTIFIER, hrDeviceNetwork}
{hrDeviceDescr.7              , OCTET STRING, ln0 - DEC LANCE Ethernet Interface}
{hrDeviceID.7                 , OBJECT IDENTIFIER, 0.0}
{hrDeviceStatus.7             , INTEGER, running}
{hrDeviceErrors.7             , Counter, 40}
{hrDeviceIndex.8              , INTEGER, 8}
{hrDeviceType.8               , OBJECT IDENTIFIER, hrDeviceNetwork}
{hrDeviceDescr.8              , OCTET STRING, sl0 - Serial Line Interface}
{hrDeviceID.8                 , OBJECT IDENTIFIER, 0.0}
{hrDeviceStatus.8             , INTEGER, down}
{hrDeviceErrors.8             , Counter, 0}
{hrDeviceIndex.9              , INTEGER, 9}
{hrDeviceType.9               , OBJECT IDENTIFIER, hrDeviceNetwork}
{hrDeviceDescr.9              , OCTET STRING, lo0 - Local Loopback Interface.}
{hrDeviceID.9                 , OBJECT IDENTIFIER, 0.0}
{hrDeviceStatus.9             , INTEGER, unknown}
{hrDeviceErrors.9             , Counter, 0}
{hrDeviceIndex.10             , INTEGER, 10}
{hrDeviceType.10              , OBJECT IDENTIFIER, hrDeviceNetwork}
{hrDeviceDescr.10             , OCTET STRING, ppp0 - 2.2}
{hrDeviceID.10                , OBJECT IDENTIFIER, 0.0}
{hrDeviceStatus.10            , INTEGER, down}
{hrDeviceErrors.10            , Counter, 0}
{hrDeviceIndex.11             , INTEGER, 11}
{hrDeviceType.11              , OBJECT IDENTIFIER, hrDeviceDiskStorage}
{hrDeviceDescr.11             , OCTET STRING, /dev/rz0 - SCSI RZ28}
{hrDeviceID.11                , OBJECT IDENTIFIER, 0.0}
{hrDeviceStatus.11            , INTEGER, running}
{hrDeviceErrors.11            , Counter, 0}
{hrDeviceIndex.12             , INTEGER, 12}
{hrDeviceType.12              , OBJECT IDENTIFIER, hrDeviceDiskStorage}
{hrDeviceDescr.12             , OCTET STRING, /dev/rz1 - SCSI RZ28}
{hrDeviceID.12                , OBJECT IDENTIFIER, 0.0}
{hrDeviceStatus.12            , INTEGER, running}
{hrDeviceErrors.12            , Counter, 0}
{hrDeviceIndex.13             , INTEGER, 13}
{hrDeviceType.13              , OBJECT IDENTIFIER, hrDeviceDiskStorage}
{hrDeviceDescr.13             , OCTET STRING, /dev/rz6 - SCSI RRD43}
{hrDeviceID.13                , OBJECT IDENTIFIER, 0.0}
{hrDeviceStatus.13            , INTEGER, down}
```

```
{hrDeviceErrors.13              , Counter, 0}
{hrProcessorFrwID.1             , OBJECT IDENTIFIER, 0.0}
{hrProcessorLoad.1              , INTEGER, 4}
{hrProcessorFrwID.2             , OBJECT IDENTIFIER, 0.0}
{hrProcessorLoad.2              , INTEGER, 0}
{hrProcessorFrwID.3             , OBJECT IDENTIFIER, 0.0}
{hrProcessorLoad.3              , INTEGER, 10}
{hrProcessorFrwID.4             , OBJECT IDENTIFIER, 0.0}
{hrProcessorLoad.4              , INTEGER, 19}
{hrDiskStorageAccess.11         , INTEGER, readWrite}
{hrDiskStorageMedia.11          , INTEGER, unknown}
{hrDiskStorageRemoveble.11      , INTEGER, false}
{hrDiskStorageCapacity.11       , INTEGER, 2055240}
{hrDiskStorageAccess.12         , INTEGER, readWrite}
{hrDiskStorageMedia.12          , INTEGER, unknown}
{hrDiskStorageRemoveble.12      , INTEGER, false}
{hrDiskStorageCapacity.12       , INTEGER, 2055240}
{hrDiskStorageAccess.13         , INTEGER, readWrite}
{hrDiskStorageMedia.13          , INTEGER, unknown}
{hrDiskStorageRemoveble.13      , INTEGER, false}
{hrDiskStorageCapacity.13       , INTEGER, 0}
{hrPartitionIndex.11.1          , INTEGER, 1}
{hrPartitionLabel.11.1          , OCTET STRING, /dev/rz0a}
{hrPartitionID.11.1             , OCTET STRING, }
{hrPartitionSize.11.1           , INTEGER, 65536}
{hrPartitionFSIndex.11.1        , INTEGER, 1}
{hrPartitionIndex.11.2          , INTEGER, 2}
{hrPartitionLabel.11.2          , OCTET STRING, /dev/rz0b}
{hrPartitionID.11.2             , OCTET STRING, }
{hrPartitionSize.11.2           , INTEGER, 200704}
{hrPartitionFSIndex.11.2        , INTEGER, 0}
{hrPartitionIndex.11.3          , INTEGER, 3}
{hrPartitionLabel.11.3          , OCTET STRING, /dev/rz0c}
{hrPartitionID.11.3             , OCTET STRING, }
{hrPartitionSize.11.3           , INTEGER, 2055240}
{hrPartitionFSIndex.11.3        , INTEGER, 0}
{hrPartitionIndex.11.4          , INTEGER, 4}
{hrPartitionLabel.11.4          , OCTET STRING, /dev/rz0d}
{hrPartitionID.11.4             , OCTET STRING, }
{hrPartitionSize.11.4           , INTEGER, 595968}
{hrPartitionFSIndex.11.4        , INTEGER, 0}
{hrPartitionIndex.11.5          , INTEGER, 5}
{hrPartitionLabel.11.5          , OCTET STRING, /dev/rz0e}
{hrPartitionID.11.5             , OCTET STRING, }
{hrPartitionSize.11.5           , INTEGER, 595968}
{hrPartitionFSIndex.11.5        , INTEGER, 0}
{hrPartitionIndex.11.6          , INTEGER, 6}
{hrPartitionLabel.11.6          , OCTET STRING, /dev/rz0f}
{hrPartitionID.11.6             , OCTET STRING, }
{hrPartitionSize.11.6           , INTEGER, 597064}
{hrPartitionFSIndex.11.6        , INTEGER, 0}
{hrPartitionIndex.11.7          , INTEGER, 7}
{hrPartitionLabel.11.7          , OCTET STRING, /dev/rz0g}
{hrPartitionID.11.7             , OCTET STRING, }
{hrPartitionSize.11.7           , INTEGER, 893952}
{hrPartitionFSIndex.11.7        , INTEGER, 3}
{hrPartitionIndex.11.8          , INTEGER, 8}
{hrPartitionLabel.11.8          , OCTET STRING, /dev/rz0h}
{hrPartitionID.11.8             , OCTET STRING, }
{hrPartitionSize.11.8           , INTEGER, 895048}
{hrPartitionFSIndex.11.8        , INTEGER, 0}
{hrPartitionIndex.12.1          , INTEGER, 1}
{hrPartitionLabel.12.1          , OCTET STRING, /dev/rz1a}
{hrPartitionID.12.1             , OCTET STRING, }
```

```
{hrPartitionSize.12.1           , INTEGER, 65536}
{hrPartitionFSIndex.12.1        , INTEGER, 0}
{hrPartitionIndex.12.2          , INTEGER, 2}
{hrPartitionLabel.12.2          , OCTET STRING, /dev/rz1b}
{hrPartitionID.12.2             , OCTET STRING, }
{hrPartitionSize.12.2           , INTEGER, 200704}
{hrPartitionFSIndex.12.2        , INTEGER, 0}
{hrPartitionIndex.12.3          , INTEGER, 3}
{hrPartitionLabel.12.3          , OCTET STRING, /dev/rz1c}
{hrPartitionID.12.3             , OCTET STRING, }
{hrPartitionSize.12.3           , INTEGER, 2055240}
{hrPartitionFSIndex.12.3        , INTEGER, 0}
{hrPartitionIndex.12.4          , INTEGER, 4}
{hrPartitionLabel.12.4          , OCTET STRING, /dev/rz1d}
{hrPartitionID.12.4             , OCTET STRING, }
{hrPartitionSize.12.4           , INTEGER, 595968}
{hrPartitionFSIndex.12.4        , INTEGER, 0}
{hrPartitionIndex.12.5          , INTEGER, 5}
{hrPartitionLabel.12.5          , OCTET STRING, /dev/rz1e}
{hrPartitionID.12.5             , OCTET STRING, }
{hrPartitionSize.12.5           , INTEGER, 595968}
{hrPartitionFSIndex.12.5        , INTEGER, 0}
{hrPartitionIndex.12.6          , INTEGER, 6}
{hrPartitionLabel.12.6          , OCTET STRING, /dev/rz1f}
{hrPartitionID.12.6             , OCTET STRING, }
{hrPartitionSize.12.6           , INTEGER, 597064}
{hrPartitionFSIndex.12.6        , INTEGER, 0}
{hrPartitionIndex.12.7          , INTEGER, 7}
{hrPartitionLabel.12.7          , OCTET STRING, /dev/rz1g}
{hrPartitionID.12.7             , OCTET STRING, }
{hrPartitionSize.12.7           , INTEGER, 893952}
{hrPartitionFSIndex.12.7        , INTEGER, 0}
{hrPartitionIndex.12.8          , INTEGER, 8}
{hrPartitionLabel.12.8          , OCTET STRING, /dev/rz1h}
{hrPartitionID.12.8             , OCTET STRING, }
{hrPartitionSize.12.8           , INTEGER, 895048}
{hrPartitionFSIndex.12.8        , INTEGER, 0}
{hrNetworkIfIndex.5             , INTEGER, 1}
{hrNetworkIfIndex.6             , INTEGER, 2}
{hrNetworkIfIndex.7             , INTEGER, 3}
{hrNetworkIfIndex.8             , INTEGER, 4}
{hrNetworkIfIndex.9             , INTEGER, 5}
{hrNetworkIfIndex.10            , INTEGER, 6}
```

F.5 File System Table

The file system table implementation is as follows:

- Each currently mounted file system is represented in the `hrFSTable` group.
- The available values for the `hrFSType` parameter do not cover all possible file system types in the operating system. Some types (for example, `/proc`) report a value of `hrFSOther` for the `hrFSType` object.
- The `hrFSRemoteMountPoint` parameter is returned as a zero-length octet string for local file systems, as specified in RFC 1514.
- The `hrFSStorageIndex` parameter returns a zero (0) for remote file systems, in accordance with RFC 1514. For local file systems,

the hrFSStorageIndex parameter returns the value of the hrStorageIndex parameter for the hrStorageEntry entry corresponding to that file system.

The RFC specifies this design, presumably so that all storage-related information is available in one table. However, in order to discover file system full conditions, an SNMP application needs to do the following:

1. Locate an entry in the the hrFSTable group.
2. Retrieve that entry's value of the hrFSStorageIndex parameter. For example, call it *i*.
3. If *i* is not zero (0), retrieve the values of the hrStorageUsed.*i* and hrStorageSize.*i* parameters.

- The value of the hrFSBootable parameter is always returned as false.
- The values of the hrFSLastFullBackupDate and hrFSLastPartial-BackupDate parameters are always returned as {January 1 year 0 time 0}, in the DateAndTime format, as specified in RFC 1514, when these values are unknown.

The following are sample file system table data:

```
{hrFSIndex.1                    , INTEGER, 1}
{hrFSMountPoint.1               , OCTET STRING, /}
{hrFSRemoteMountPoint.1         , OCTET STRING, }
{hrFSType.1                     , OBJECT IDENTIFIER, hrFSBerkeleyFFS}
{hrFSAccess.1                   , INTEGER, readWrite}
{hrFSBootable.1                 , INTEGER, false}
{hrFSStorageIndex.1             , INTEGER, 99}
{hrFSLastFullBackupDate.1       , OCTET STRING, 0-1-1,0:0:0.0}
{hrFSLastPartialBackupDate.1    , OCTET STRING, 0-1-1,0:0:0.0}
{hrFSIndex.2                    , INTEGER, 2}
{hrFSMountPoint.2               , OCTET STRING, /proc}
{hrFSRemoteMountPoint.2         , OCTET STRING, }
{hrFSType.2                     , OBJECT IDENTIFIER, hrFSOther}
{hrFSAccess.2                   , INTEGER, readWrite}
{hrFSBootable.2                 , INTEGER, false}
{hrFSStorageIndex.2             , INTEGER, 100}
{hrFSLastFullBackupDate.2       , OCTET STRING, 0-1-1,0:0:0.0}
{hrFSLastPartialBackupDate.2    , OCTET STRING, 0-1-1,0:0:0.0}
{hrFSIndex.3                    , INTEGER, 3}
{hrFSMountPoint.3               , OCTET STRING, /usr}
{hrFSRemoteMountPoint.3         , OCTET STRING, }
{hrFSType.3                     , OBJECT IDENTIFIER, hrFSBerkeleyFFS}
{hrFSAccess.3                   , INTEGER, readWrite}
{hrFSBootable.3                 , INTEGER, false}
{hrFSStorageIndex.3             , INTEGER, 101}
{hrFSLastFullBackupDate.3       , OCTET STRING, 0-1-1,0:0:0.0}
{hrFSLastPartialBackupDate.3    , OCTET STRING, 0-1-1,0:0:0.0}
{hrFSIndex.4                    , INTEGER, 4}
{hrFSMountPoint.4               , OCTET STRING, /tools}
{hrFSRemoteMountPoint.4         , OCTET STRING, /tools@tools}
{hrFSType.4                     , OBJECT IDENTIFIER, hrFSNFS}
{hrFSAccess.4                   , INTEGER, readWrite}
{hrFSBootable.4                 , INTEGER, false}
{hrFSStorageIndex.4             , INTEGER, 0}
{hrFSLastFullBackupDate.4       , OCTET STRING, 0-1-1,0:0:0.0}
```

```
{hrFSLastPartialBackupDate.4    , OCTET STRING, 0-1-1,0:0:0.0}
```

F.6 Running Software Tables

The running software table implementation is as follows:

- The `hrSWOSIndex` parameter is always returned as zero (0), the kernel idle process. There is no one process that represents the primary operating system running on this host for Tru64 UNIX.
- Each process is represented as an entry in both the `hrSWRunTable` group and the `hrSWRunPerfTable` group. The value of the `hrSWRunIndex` parameter (used to index both tables) is the pid of that process. This means there is an entry whose `hrSWRunIndex` parameter value is 0 (zero), which is not typical of SNMP tables.
- The `hrSWRunName` parameter is always returned as a zero-length octet string.
- The `hrSWRunType` parameter is always returned as `unknown`.
- The `hrSWRunStatus` parameter is returned as either `running` (processes that are capable of being run or are waiting for CPU), or `notrunnable` (stopped or waiting for non-CPU resources).
- The `hrSWRunPath` parameter and the `hrSWRunParameters` parameter return the command and parameters, respectively, that were used to start this process. This is similar, but not identical, to the output of the `ps` command.
- The `hrSWRunPerfCPU` parameter returns the sum of accumulated system and user time for all threads running in a process. This value is equivalent to the value returned by the `ps` cputime specifier (adjusted to units of centiseconds).
- The `hrSWRunPerfMem` parameter returns the current resident set size of the process. This value is equivalent to the value returned by the `ps rssize` specifier, adjusted to units of 1024 bytes (`Kbytes` are defined in RFC 1514).

The following are sample running software table data:

```
{hrSWRunIndex.0                 , INTEGER, 0}
{hrSWRunName.0                  , OCTET STRING, }
{hrSWRunID.0                    , OBJECT IDENTIFIER, 0.0}
{hrSWRunPath.0                  , OCTET STRING, }
{hrSWRunParameters.0            , OCTET STRING, }
{hrSWRunType.0                  , INTEGER, unknown}
{hrSWRunStatus.0                , INTEGER, running}
{hrSWRunIndex.1                 , INTEGER, 1}
{hrSWRunName.1                  , OCTET STRING, }
{hrSWRunID.1                    , OBJECT IDENTIFIER, 0.0}
{hrSWRunPath.1                  , OCTET STRING, /sbin/init}
{hrSWRunParameters.1            , OCTET STRING,  -a}
{hrSWRunType.1                  , INTEGER, unknown}
```

```
{hrSWRunStatus.1                , INTEGER, notRunnable}
{hrSWRunIndex.3                 , INTEGER, 3}
{hrSWRunName.3                  , OCTET STRING, }
{hrSWRunID.3                    , OBJECT IDENTIFIER, 0.0}
{hrSWRunPath.3                  , OCTET STRING, /sbin/kloadsrv}
{hrSWRunParameters.3            , OCTET STRING, }
{hrSWRunType.3                  , INTEGER, unknown}
{hrSWRunStatus.3                , INTEGER, notRunnable}
{hrSWRunIndex.16                , INTEGER, 16}
{hrSWRunName.16                 , OCTET STRING, }
{hrSWRunID.16                   , OBJECT IDENTIFIER, 0.0}
{hrSWRunPath.16                 , OCTET STRING, /sbin/update}
{hrSWRunParameters.16           , OCTET STRING, }
{hrSWRunType.16                 , INTEGER, unknown}
{hrSWRunStatus.16               , INTEGER, notRunnable}
   .
   .
{hrSWRunIndex.142               , INTEGER, 142}
{hrSWRunName.142                , OCTET STRING, }
{hrSWRunID.142                  , OBJECT IDENTIFIER, 0.0}
{hrSWRunPath.142                , OCTET STRING, /usr/sbin/routed}
{hrSWRunParameters.142          , OCTET STRING,  -q}
{hrSWRunType.142                , INTEGER, unknown}
{hrSWRunStatus.142              , INTEGER, notRunnable}
{hrSWRunIndex.228               , INTEGER, 228}
{hrSWRunName.228                , OCTET STRING, }
{hrSWRunID.228                  , OBJECT IDENTIFIER, 0.0}
{hrSWRunPath.228                , OCTET STRING, /usr/sbin/nfsiod}
{hrSWRunParameters.228          , OCTET STRING,  7}
{hrSWRunType.228                , INTEGER, unknown}
{hrSWRunStatus.228              , INTEGER, notRunnable}
{hrSWRunIndex.394               , INTEGER, 394}
{hrSWRunName.394                , OCTET STRING, }
{hrSWRunID.394                  , OBJECT IDENTIFIER, 0.0}
{hrSWRunPath.394                , OCTET STRING, /usr/dt/bin/dtlogin}
{hrSWRunParameters.394          , OCTET STRING,  -daemon}
{hrSWRunType.394                , INTEGER, unknown}
{hrSWRunStatus.394              , INTEGER, notRunnable}
{hrSWRunIndex.395               , INTEGER, 395}
{hrSWRunName.395                , OCTET STRING, }
{hrSWRunID.395                  , OBJECT IDENTIFIER, 0.0}
{hrSWRunPath.395                , OCTET STRING, /usr/sbin/getty}
{hrSWRunParameters.395          , OCTET STRING,  console console vt100}
{hrSWRunType.395                , INTEGER, unknown}
{hrSWRunStatus.395              , INTEGER, notRunnable}
{hrSWRunIndex.396               , INTEGER, 396}
{hrSWRunName.396                , OCTET STRING, }
{hrSWRunID.396                  , OBJECT IDENTIFIER, 0.0}
{hrSWRunPath.396                , OCTET STRING, /usr/bin/X11/X}
{hrSWRunParameters.396          , OCTET STRING,  :0 -auth /var/dt/A:0-aaamka}
{hrSWRunType.396                , INTEGER, unknown}
{hrSWRunStatus.396              , INTEGER, notRunnable}
{hrSWRunIndex.397               , INTEGER, 397}
{hrSWRunName.397                , OCTET STRING, }
{hrSWRunID.397                  , OBJECT IDENTIFIER, 0.0}
{hrSWRunPath.397                , OCTET STRING, dtlogin}
{hrSWRunParameters.397          , OCTET STRING,  <:0> -daemon}
{hrSWRunType.397                , INTEGER, unknown}
{hrSWRunStatus.397              , INTEGER, notRunnable}
   .
   .
{hrSWRunPerfCPU.0               , INTEGER, 9288}
{hrSWRunPerfMem.0               , INTEGER, 10024}
{hrSWRunPerfCPU.1               , INTEGER, 34}
{hrSWRunPerfMem.1               , INTEGER, 64}
```

```
{hrSWRunPerfCPU.3               , INTEGER, 17}
{hrSWRunPerfMem.3               , INTEGER, 2000}
{hrSWRunPerfCPU.16              , INTEGER, 4476}
{hrSWRunPerfMem.16              , INTEGER, 88}
   .
   .
   .
{hrSWRunPerfCPU.142             , INTEGER, 891}
{hrSWRunPerfMem.142             , INTEGER, 112}
{hrSWRunPerfCPU.228             , INTEGER, 0}
{hrSWRunPerfMem.228             , INTEGER, 56}
{hrSWRunPerfCPU.394             , INTEGER, 51}
{hrSWRunPerfMem.394             , INTEGER, 264}
{hrSWRunPerfCPU.395             , INTEGER, 7}
{hrSWRunPerfMem.395             , INTEGER, 80}
{hrSWRunPerfCPU.396             , INTEGER, 4329}
{hrSWRunPerfMem.396             , INTEGER, 2648}
{hrSWRunPerfCPU.397             , INTEGER, 8}
{hrSWRunPerfMem.397             , INTEGER, 232}
   .
   .
   .
```

G

Format of DNS Data File Entries

The Domain Name System (DNS) configuration file, by default called `/etc/namedb/named.conf`, specifies the names of the DNS data files. These data files consist of entries, also known as Resource Records (RR), that follow the formats described in this chapter.

G.1 Format of DNS Resource Records

Here is the general format of a DNS Resource Record:

name ttl addr-class entry-type entry-specific-data

The fields are defined as follows:

Field	Description
name	This is the name of the domain, for example `cities.dec.com`. The domain name must begin in the first column. For some data file entries the name field is left blank. In that case, the domain name is assumed to be the same as the previous entry. A free standing period (.) refers to the current domain. A free standing at sign (@) denotes the current origin, thus allowing you to specify more than one domain. Two free standing periods (..) represent the null domain name of the root.
ttl	This is the time-to-live field, and specifies how long, in seconds, the data will be stored in the database. If this field is left blank, the value defaults to the ttl value specified in the SOA (start of authority) entry or, ultimately, the value of the $ttl entry. The maximum time-to-live is 99999999 seconds, or 3 years.

Field	Description
addr-class	This field is the address class. There are three classes: IN — Internet addresses, TXT — naming service data, ANY — all other types of network addresses. The address class of all data file entries of a given entry-type in a particular zone must be the same. Therefore, only the first entry in a zone need specify the *addr-class* field.
entry-type	This field states the resource record type, for example SOA (start of authority) or A (address).
entry-specific-data	All fields after the entry-type field vary for each type of date file entry (resource record).

The case is preserved in name and data fields when loaded into the DNS server. Comparisons and lookups using DNS are case insensitive.

The following characters have special meanings in DNS data file entries:

Character	Meaning
x	A backslash (\) escapes the next nondigit (*x*) character so that the character's special meaning does not apply. For example, you could use a period (.) to place a period character in a label.
nnn	A backslash denotes the octet corresponding to the decimal number represented by *nnn*. The resulting octet is assumed to be text and is not checked for special meaning.
()	Parentheses group data that cross a line. In effect, line terminations are not recognized within parentheses.
;	A semicolon starts a comment, causing the rest of the line to be ignored.
*	An asterisk signifies a wildcard.

Most DNS data file entries have the current domain appended to their names if they are not terminated by a period (.). This is useful for appending the current domain name to the data, such as system names, but could cause problems when you do not want this to happen. Therefore, if the name is not in the domain for which you are creating the data file, end the name with a period.

Data files (resource records) can have the following types of entries:

- $include
- $origin

- $ttl — time to live
- A — address
- AAAA — IPv6 address
- CNAME — canonical name
- HINFO — host information
- MB — mail box
- MG — mail group
- MINFO — mailbox information
- MR — mail rename
- MX — mail exchanger
- NS — name server
- PTR — domain name pointer
- SOA — start of authority
- SRV — location of services
- WKS — well known services

G.2 Description of Data File Entries

The following sections describe each data file entry and its format.

G.2.1 Include Entry

An include entry is similar to a header file in the C programming language. This feature is particularly useful for separating different types of data into multiple files. An include entry begins with `$include` in the first column, and is followed by the name of the file to be included. For example:

```
$include /etc/namedb/mailboxes
```

This entry requests DNS to load the data file `/etc/namedb/mailboxes`.

The include entry loads data files into the local zone and acts as a data file organizer. For example, you can use `$include` entries to separate mail from host information.

G.2.2 Origin Entry

An origin entry changes the origin in a data file. This feature is particularly useful for putting more than one domain in a data file. An origin entry begins with `$origin` in the first column, followed by a domain origin, as shown in the following example:

```
$origin state.dec.com.
```

This entry includes the domain `state.dec.com` in the data file. As a result, DNS can provide information about the `state.dec.com` domain in addition to the local domain, provided your server has authority for the zone.

The `$origin` and `$include` entries can work together. They can save typing and help keep the files organized. For example, assume that the following entries are in the `hosts.rev` file:

```
$origin ll.128.in-addr.arpa.
$include cities.dec.com.rev
```

The period after `arpa` signifies the complete domain name. Assume that the `cities.dec.com.rev` file consists of entries similar to the following:

```
33.22 IN PTR chicago.cities.dec.com.
```

In this situation, the complete reverse name for the host chicago is translated to be as follows:

```
33.22.11.128. in-addr.arpa. IN PTR chicago.cities.dec.com.
```

G.2.3 TTL Entry

The time-to-live entry is similar to the ttl field in other resource records; it specifies how long data will be stored in the cache. However, when you set the time-to-live in the optional `$ttl` entry, the limit takes takes effect only if no time-to-live value is specified for a particular resource record or its corresponding SOA record.

A `$ttl` entry begins with `$ttl` in the first column, a value in the second column, and an optional comment in the third column. For example, this entry specifies that resource records without a specified ttl will expire after 21600 seconds (or six hours):

```
$ttl 21600 default time to live
```

When you specify it in this manner, the time-to-live value must be in the range of 0 to 2147483647 seconds. Alternatively, you can specify the time-to-live in the following format, where you need not specify all of the fields:

*weeks*W*days*D*hours*H*minutes*M*seconds*S

For example, the maximum value in this format (3550 weeks, 5 days, 3 hours, 14 minutes, 7 seconds) would be specified as follows:

```
$ttl 3550W5D3H14M7S
```

G.2.4 Address Entry

The address (A) data file entry lists the IPv4 address for a specific system. An A entry has the following format:

name ttl addr-class entry-type address

The fields in the A entry have the values described in Section G.1, with the exception of the *address* field. This field specifies the IPv4 address for each system. There must be only one A entry for each address on a given system.

The following is an example of two A entries:

```
;name                  ttl    addr-class    entry-type    address
miami.cities.dec.com.         IN            A             128.11.22.44
                              IN            A             128.11.22.33
```

In this example, the host miami.cities.dec.com has two IP addresses, both IPv4. (See Section G.2.5 for an example of a host with an IPv4 address and an IPv6 address.)

Note that in the first entry the *ttl* field is blank, thus using the default ttl specified in the SOA entry or the $ttl entry. In the second entry, the first and second fields are blank, thus using the default name specified in the previous entry and the same default ttl.

G.2.5 IPv6 Address Entry

The IPv6 address (AAAA) data file entry lists the address for a specific IPv6 system. An AAAA entry has the following format:

name ttl addr-class entry-type address

The fields in the AAAA entry have the values described in Section G.1, with the exception of the *address* field. This field specifies the IPv6 address for each system. There must be only one AAAA entry for each address on a given system.

The following is an example of a AAAA entry:

```
;name                  ttl    addr-class    entry-type    address
boston.cities.dec.com.          IN              A         128.11.22.42
                                IN             AAAA      1070:0:0:0:0:800:200C:417B
```

In this example, the host boston.cities.dec.com has two IP addresses, one IPv4 address and one IPv6 address. The coupling of the AAAA entry with the A entry is typical of some IPv6 configurations because it provides backwards compatibility for IPv4 networks.

See *Network Administration: Connections* for more information about IPv6.

G.2.6 Canonical Name Entry

The canonical name (CNAME) entry specifies an alias for a canonical name. For example, if the canonical name (also known as the full DNS name or the fully qualified name) is miami.cities.dec.com, a reasonable alias might be miami or mi.

An alias must be unique, and all other entries must be associated with the canonical name and not with the alias. Do not create an alias and then use it in other entries. A CNAME entry has the following format:

aliases ttl addr-class entry-type can-name

The fields in the CNAME entry have the values described in Section G.1, with the following exceptions:

Field	Description
alias	This field specifies the nickname (alias) of the canonical name of the host.
can-name	This is the canonical name of the host. If the canonical name is a part of the current domain, you need to specify only the host name; for example, miami. If the canonical name is for a host in another domain, you must specify the fully qualified DNS name, followed by a period (.). For example: ohio.state.dec.com.

The following example shows two CNAME entries. The first entry is for a CNAME in the current domain, cities.dec.com; the second entry is for a CNAME in another domain:

```
;aliases    ttl    addr-class    entry-type    can-name
to                 IN            CNAME         toledo
oh                 IN            CNAME         ohio.state.dec.com.
```

G.2.7 Host Information Entry

The host information (HINFO) data file entry is for host specific information. This entry lists the hardware and operating system that are running at the specified host system. Only a single space separates the name of the hardware from the operating system information. Thus, if you need to use spaces as part of a host or operating system name, you must place the name in quotation marks. In addition, there can be no more than one HINFO entry for each host on the domain. The following is the HINFO entry format:

host ttl addr-class entry-type hardware opsys

The fields in the HINFO entry have the values described in Section G.1, with the following exceptions:

Field	Description
host	This field specifies the host name. If the host is in the current domain, you need to specify only the host; for example, chicago. If the host is in a different domain, you must specify the full DNS name, for example, utah.state.dec.com.. Be sure to include the period (.) at the end of the host name. This indicates the fully qualified DNS name.
hardware	This field specifies the type of CPU; for example, an AlphaServer 8400.
opsys	This field specifies the type of operating system running on the specified host. Its recommended setting is Tru64 UNIX for the Tru64 UNIX operating system.

The following is an example of a HINFO entry:

```
;name               ttl   addr-class   entry-type   hardware            opsys
ohio.state.dec.com.       IN           HINFO        "AlphaServer
8400"   "Tru64 UNIX"
```

In this example, note that the second field specifying the ttl is blank, thus using the default ttl specified in the SOA entry or the $ttl entry.

G.2.8 Mailbox Entry

The mailbox (MB) entry lists the system where a user wants to receive mail. The following is the format of an MB entry:

login ttl addr-class entry-type system

The fields in the MB entry have the values described in Section G.1, with the following exceptions:

Field	Description
login	This field is the login name for a user. Login names must be unique for the domain.
system	This field specifies the name system where the user wants to receive mail.

The following is an example of an MB entry:

```
;login    ttl    addr-class    entry-type    system
fred              IN            MB            potsdam.cities.dec.com.
```

In this example, note that the second field is blank, thus using the default ttl specified in the SOA entry or the $ttl entry. Consequently, the user Fred will have mail delivered to the host named potsdam in the domain cities.dec.com.

G.2.9 Mail Group Entry

The mail group (MG) entry specifies the members of a mail group. The MG entry is usually used with a MINFO entry. The following is the format of an MG entry:

group ttl addr-class entry-type member

The fields in the MG entry have the values described in Section G.1, with the following exceptions:

Field	Description
group	This field specifies the name of the mail group, for example, users or marketing.
member	This field specifies the login name and the domain of the user to be included in the mail group.

The following is an example of a MINFO entry and several MG entries:

```
;group    ttl    addr-class    entry-type    requests        member
fun              IN            MINFO         BIND-REQUEST    fred@miami.cities.dec.com.
                 IN            MG                            john@miami.cities.dec.com.
                               MG
amy@miami.cities.dec.com.
```

In this example, note that the second field for all three entries is blank, thus using the default ttl specified in the SOA entry or the $ttl entry. In addition, Fred, John, and Amy will receive any mail sent to the mail group fun.

G.2.10 Mailbox Information Entry

The mailbox information (MINFO) entry creates a mail group for a mailing list. The MINFO entry is usually associated with a mail group (MG) entry, but can also be used with a mailbox (MB) entry. The following is the format of a MINFO entry:

mailbox ttl addr-class entry-type requests maintainer

The fields in the MINFO entry have the values described in Section G.1, with the following exceptions:

Field	Description
mailbox	This field specifies the name of the mailbox, and its value is usually `BIND`.

Field	Description
requests	This field specifies the name where users can send mail relating to DNS or mail. For example, a user might want to send a mail message requesting that an alias be set up.
maintainer	This field contains the login name of the person who will receive mail error messages. This is particularly useful when an error in member's names needs to be reported to a person other than the sender.

The following is an example of a MINFO entry:

```
mailbox    ttl    addr-class    entry-type    requests        maintainer
BIND              IN            MINFO         BIND-REQUEST
fred@miami.cities.dec.com.
```

In this example, note that the second field is blank, thus using the default ttl specified in the SOA entry or the $ttl entry.

G.2.11 Mail Rename Entry

The mail rename (MR) entry lists aliases for a specific user. The following is the format of an MR entry:

alias ttl addr-class entry-type login

The fields in the MR entry have the values described in Section G.1, with the following exceptions:

Field	Description
alias	This field lists the nicknames for the specified user. The alias must be unique to the domain.
login	This field is the login name for the user whose alias is being established. There should also be a corresponding MB entry for the specified login name. Login names must be unique for the domain.

The following is an example of an MR entry:

```
;alias    ttl    addr-class    entry-type    login
lady              IN            MR            diana
princess          IN            MR            diana
```

This example shows how to set up the aliases lady and princess for a user whose login name is diana. Note that the second field is left blank, thus using the default ttl specified in the SOA entry or the $ttl entry.

G.2.12 Mail Exchanger Entry

The mail exchanger (MX) entry specifies a system in the local domain (called a gateway) that knows how to deliver mail to a system that might not be directly connected to the local network. Consequently, the MX entry is useful for systems outside your local network that want to send mail to a user on one of your network's hosts.

You can also use the MX entry to list some of the hosts in the `/etc/hosts` file so that they do not appear to other systems using DNS service.

The following is the format of an MX entry:

system ttl addr-class entry-type pref-value gateway

The fields in the MX entry have the values described in Section G.1, with the following exceptions:

Field	Description
system	This field specifies the name of the system where mail is to be sent.
pref-value	This field specifies the order a mailer is to follow when there is more than one way to deliver mail to a given system.
gateway	This field contains the name of the gateway system, that is, the system that can deliver mail to the destination system on another network.

The following is an example of two MX entries:

```
;system                ttl    addr-class    entry-type    pref-value    gateway
tampa.cities.dec.com           IN            MX            0             seismo.cs.au.
*.folks.dec.com                IN            MX            0             relay.cs.net.
```

In this example, all mail destined for the domain folks.dec.com, regardless of the host name, is sent by route of the relay.cs.net host. In addition, note that the ttl field in both entries is blank, thus using the default ttl specified in the SOA entry or the $ttl entry. The second entry uses an asterisk, which is a wildcard.

G.2.13 Name Server Entry

The name server (NS) entry specifies that a system is a name server for the specified domain. The following is the format of the NS entry:

name ttl addr-class entry-type server

The fields in the NS entry have the values described in Section G.1, with the exception of the *server* field. This field specifies the name of the primary master server for the domain specified in the first field.

The following is an example of an NS entry:

```
;name    ttl    addr-class    entry-type    server
                IN            NS            utah.states.dec.com.
```

G.2.14 Domain Name Pointer Entry

The domain name pointer (PTR) entry allows special names to point to some other location in the domain. PTR names must be unique to the zone. These entries are located on a primary server in the `/etc/namedb/hosts.rev` file. The following is the format of a PTR entry:

rev-addr ttl addr-class entry-type hostname

The fields in the PTR entry have the values described in Section G.1, with the following exceptions:

Field	Description
rev-addr	This field specifies the reverse IP address of the host. For example, if the host's address is 128.11.22.33, the reverse address is 33.22.11.128.in-addr.arpa. This field also supports IPv6 addresses. An IPv6 address is represented in reverse order as a sequence of 4–bit nibbles separated by dots with the suffix `.IP6.INT`.
hostname	This is the fully qualified DNS name of the host, for example, miami.cities.dec.com. Be sure to include the period (.) at the end of the host name if the host is not in the current domain.

The following is an example of two IPv4 PTR entries:

```
;rev-addr                    ttl    addr-class     entry-type    hostname
33.22                               IN             PTR           chicago
66.55.44.121.in-addr.arpa.          IN             PTR           mail.peace.org.
```

In this example, the first entry is for a host whose IP host address is 22.33 in the current domain. The specified reverse address (33.22) is meaningful assuming that a `$origin` entry exists. See Section G.2.2 for a description of the `$origin` entry. If there is not an `$origin` entry, then the entire IP address, in reverse, must be specified.

The second entry is for a host in different domain (mail.peace.org.). As a rule, do not do this because you are putting data in your server's cache for which your server is not authoritative. PTR entries and other resource records are for hosts in your domain only. The PTR entry sets up a reverse pointer for the host mail.peace.org.

The following is an example of an IPv6 PTR entry:

```
;rev-addr         ttl    addr-class     entry-type    hostname
$ORIGIN 0.0.7.a.f.e.f.f.f.8.f.0.0.2.0.0.6.8.1.1.0.2.0.0.5.0.8.e.f.f.3.ip6.int.
d                3600   IN             PTR           equinox.ipv6.campus.edu.
```

In this example, the administrator uses the `$origin` entry to create a more organized resource record. By specifying most of the IPv6 address in the `$origin` entry, you can prevent the PTR entry from wrapping over to the next line.

The forward IPv6 address for this entry is 3ffe:8050:201:1860:0200:f8ff:fefa:700d.

See *Network Administration: Connections* for more information about IPv6.

G.2.15 Start of Authority Entry

The start of authority (SOA) entry designates the beginning of a zone. There can be no more than one SOA entry per zone. The following is the format of an SOA entry:

name ttl addr-class entry-type origin person serial#
refresh retry expire min

The fields in the SOA entry have the values described in Section G.1, with the following exceptions:

Field	Description
origin	This field is the name of the host on which the data file resides. This is usually a master server.
person	This field defines the login name and mailing address of the person responsible for DNS running on the local domain.
serial#	This field specifies the version number of the data file. The person editing the master files for the zone must increment the value in this field each time a change is made to the data within the file. The serial number being changed informs the secondary servers that there is new data to be obtained from the master server. The maximum number is 2^{32}–1 after the decimal point. The serial number field allows DNS to determine which of two copies of data files in a zone are more recent. Typically, the serial number field begins at one (1) and is incremented by one each time the original data file is modified. It is best to use whole integers.

Field	Description
refresh	This field specifies how often, in seconds, a secondary DNS server is to check with the master server to see if it needs to update its data files. If the data files are out of date (as indicated by a mismatch of serial number fields), they are updated with the contents of the master server's files. The minimum refresh period is 30 seconds. If the refresh field is blank, however, the data file is not dynamically updated.
retry	This field specifies how often, in seconds, a secondary DNS server will try to refresh its data files after a refresh failure has occurred while making the check. If a DNS server attempts to refresh the files and fails, it tries to refresh them again every so many seconds, as specified in the retry field.
expire	This field specifies the upper limit, in seconds, that a secondary DNS server can use the data files in its cache before the data expires for lack of being updated, or before DNS server checks to see if its cache needs to be updated.
min	This field specifies the default time to live, in seconds, that a data entry can exist in the event that the ttl entry is left blank.

The following is an example of an SOA entry. The first line is a comment that shows the fields:

```
;name   ttl   addr-class   entry-type   origin                              person
@             IN           SOA          utah.states.dec.com. hes.utah.states.dec.com. (
                                              1       ; serial
                                              3600    ; refresh every hr.
                                              300     ; retry every 5 min.
                                              3600000 ; expire in 1000 hrs.
                                              86400 ) ; min. life is 24 hrs.
```

In this example note that the parentheses indicate to DNS that this is a single entry. The ttl field is blank, indicating that the default time to live specified in the min field (86400 seconds) is being used.

The semicolons allow comments for readability. In the example, the serial field is 1, the refresh field is 3600 seconds (once per hour), the retry field is 300 seconds (once per 5 minutes), the expire field is 3,600,000 seconds (1000 hours), and the min field is 86400 seconds (24 hours).

G.2.16 Location of Services Entry

The location of services (SRV) entry describes services supported in a particular target domain. It replaces the well-known services (WKS) entry, which is maintained for backwards compatibility. The following is the format of an SRV entry:

_service._protocol.name ttl addr-class entry-type
priority weight target

The fields in the SRV entry have the values described in Section G.1, with the following exceptions:

Field	Description
service	This case-insensitive field specifies the symbolic name of the desired service. You can select a service name from the `/etc/services` file, or define a new, unique service name locally. You must prepend an underscore (_) to the service, for example, `_ldap`, to prevent collisions with other DNS labels.
protocol	This case-insensitive field specifies the symbolic name of the protocol to use. You must prepend an underscore (_) to the protocol, for example, `_TCP` or `_UDP`, to prevent collisions with other DNS labels.
name	This field specifies the domain to which this SRV record refers. As with other resource records, the name field can be left blank if the SRV record is inserted directly below records from the same domain.
priority	This field specifies the priority, a value from 0 to 65535, of the target system or server. Clients for the specified service attempt to contact the system with the lowest priority value. Systems with the same priority are tried in an order defined by the *weight* field.
weight	This field specifies the relative weight, a value from 0 to 65535, for entries that have the same priority value. Higher weight values are more likely to be selected.

Field	Description
port	This field specifies the port of the service, a value from 0–65535, on the target server. You can obtain the port number from the `/etc/services` file, or define a new, unique port number locally.
target	This is the fully qualified host name of the target system or server, for example, miami.cities.dec.com. One or more address records must exist for this host name, and the name must not be an alias. If you need to indicate that a service is definitely not available at this domain, specify a dot (.) for the target value.

The following is an example of SRV entries:

```
;_service._protocol.name   entry   priority   weight   port   target
_ldap._tcp                 SRV     0          1        389    whitepages2.crimson.com.
                           SRV     0          3        389    whitepages.crimson.com.
                           SRV     1          0        389    zeus.crimson.com.
```

In this example, the three SRV entries describe support for the Lightweight Directory Access Protocol (LDAP) in the crimson.com domain. The primary LDAP servers for the domain are whitepages and whitepages2. They share the same priority, but because whitepages is a newer and faster server, the administrator assigns a higher weight value to it. As a result, whitepages handles a majority of the directory queries for the domain.

If both primary LDAP servers are unavailable, clients switch to querying the zeus server, which provides backup for various services in the domain.

G.2.17 Well Known Services Entry

The well known services entry (WKS) has been replaced by the Location of Services entry (SRV); however, WKS is maintained for backwards compatibility.

The WKS entry describes well known services supported by a particular protocol at a specified address. The services and port numbers are obtained from the list of services specified in the `/etc/services` file. The following is the format of a WKS entry:

name ttl addr-class entry-type address protocol services

The fields in the WKS entry have the values described in Section G.1, with the following exceptions:

Field	Description
address	This field specifies the IP address for each system. There can be only one WKS entry for each protocol at each address.
protocol	This field specifies the protocol to be used, for example TCP or UDP.

The following is an example of two WKS entries:

```
;name    ttl    addr-class    entry-type    address       protocol    services
               IN            WKS           128.32.0.4    UDP         who route
               IN            WKS           128.32.0.78   TCP         (echo talk
                                                               discard sunrpc sftp
                                                               uucp-path netstat host
                                                               systat daytime link
                                                               auth time ftp
                                                               nntp whois pop
                                                               finger smtp supdup
                                                               domain nameserver
                                                               chargen)
```

Note that the first and second fields of both entries in this example are blank, which indicates that they are using the domain name specified in a previous entry and the default ttl specified in the SOA entry or the $ttl entry. The services listed in the second entry are contained within parentheses and are thus interpreted as being one entry, even though they appear on several lines.

Index

A

B

C

D

E

F

G

H

I

J

L

M

N

P

Q

R

S

T

U

V

W

X

Y