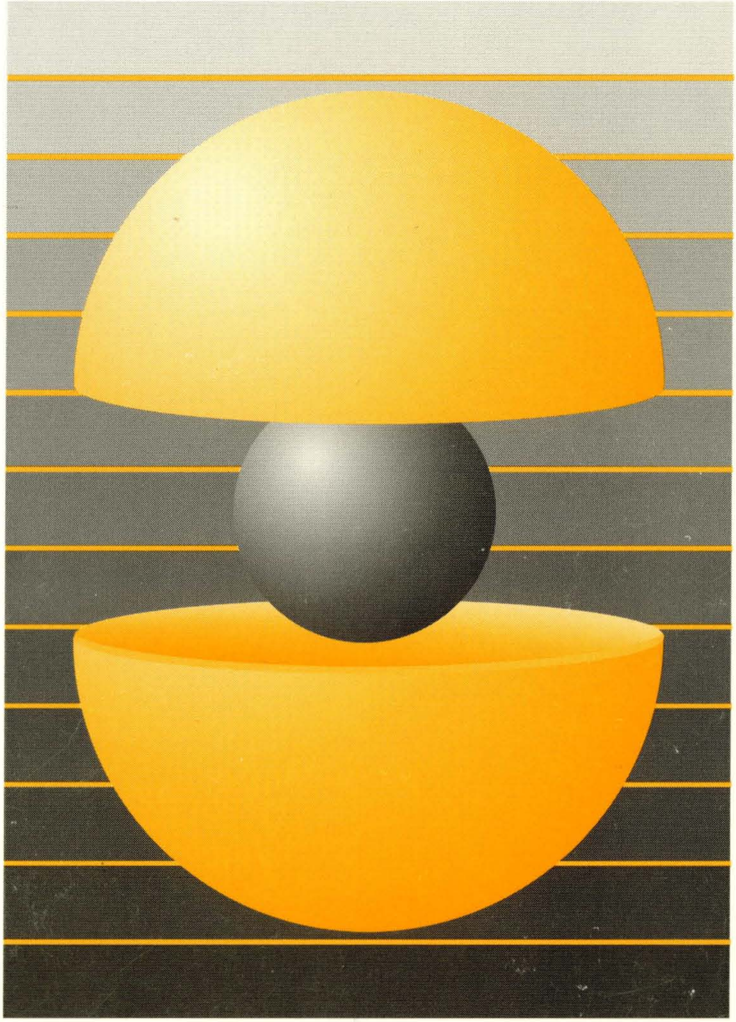
DEC OSF/1

Network Configuration



Part Number: AA-PS2GB-TE

DEC OSF/1

Network Configuration

Order Number: AA-PS2GB-TE

February 1994

Product Version: DEC OSF/1 Version 2.0 or higher

This guide describes the tasks you need to complete to establish your system on a network. This guide is intended for experienced system or network administrators.

digital equipment corporation
Maynard, Massachusetts

Contents

About This Manual

1 Overview of Network Configuration

Part 1: Suggested Setup Methods

2 Setting Up the Network

4 Setting Up the Berkeley Internet Name Domain Service

5 Setting Up the Network Information Service

6 Setting Up the Network File System

7 Setting Up the UNIX-to-UNIX Copy Program

8 Setting Up the Network Time Protocol

9 Setting Up Your Mail System

10 Setting Up the Simple Network Management Protocol Agent

Part 2: Alternative Setup Methods

11 Manually Setting Up the Network

12 Setting Up the Serial Line Internet Protocol

13 Manually Setting Up the Local Area Transport

14 Manually Setting Up the Berkeley Internet Name Domain Service

15 Manually Setting Up the Network Information Service

16 Manually Setting Up the Network File System

17 Manually Setting Up the UNIX-to-UNIX Copy Program

18 Manually Setting Up the Network Time Protocol

19 Manually Setting Up the Mail System

20 Manually Setting Up the SNMP Agent

A Configuration Worksheet

B Setting Up the Database Services Selection File

C Additional Information on the Local Area Transport

D Writing automount Maps

Examples

Figures

About This Manual

This guide provides information on the tasks you need to complete to establish your system on a network and to configure your network software (such as NFS and BIND).

Audience

This guide is for experienced system and network administrators who have knowledge of Transmission Control Protocol/Internet Protocol (TCP/IP) networking concepts and network configuration; they should also have knowledge of operating system concepts, commands, and configuration.

Organization

This guide is divided into two parts and three appendixes. Part 1 (Chapters 2 through 10) describes the suggested method for completing a task, usually using a setup script. Part 2 (Chapters 11 through 20) describes how to complete manually the tasks for which a setup script was described in Part 1.

The following list provides a brief description of the contents of each chapter:

Chapter 1 Provides an overview of the tasks described in this manual. It indicates whether a setup script is available to complete the task, and what (if any) optional software subsets must be installed to complete the task.

Chapter 2 Describes how to set up your network and get it running, using the `netsetup` utility.

Chapter 3 Describes how to set up Local Area Transport (LAT) devices, using the `latsetup` utility.

Chapter 4 Describes how to set up a Berkeley Internet Name Domain (BIND) service, using the `bindsetup` script.

Chapter 5 Describes how to set up a Network Information Service (NIS) domain, using the `nissetup` script.

Chapter 6 Describes how to set up Network File System (NFS), using the `nfssetup` script.

Chapter 7 Describes how to set up the UNIX-to-UNIX Copy Program (UUCP), using the `uucpsetup` script.

Chapter 8 Describes how to set up the Network Time Protocol (NTP), using the `ntpsetup` script.

Chapter 9 Describes how to set up and start your mail system, using the `mailsetup` script.

Chapter 10 Describes how to set up the Simple Network Management Protocol (SNMP) Agent, using the `snmpsetup` script.

Chapter 11 Describes how to set up your network manually.

Chapter 12 Describes how to set up Serial Line Internet Protocol (SLIP).

Chapter 13 Describes how to set up Local Area Transport (LAT) devices manually.

Chapter 14 Describes how to set up the Berkeley Internet Name Domain (BIND) service manually.

Chapter 15 Describes how to set up the Network Information Service (NIS) manually.

Chapter 16 Describes how to set up Network File System (NFS) manually.

Chapter 17 Describes how to set up the UNIX-to-UNIX Copy Program (UUCP) manually.

Chapter 18 Describes how to set up the Network Time Protocol (NTP) manually.

Chapter 19 Describes how to set up mail manually.

Chapter 20 Describes how to set up the Simple Network Management Protocol (SNMP) Agent manually.

Appendix A Contains the Configuration Worksheet. Copy this appendix and fill in the information before completing the tasks described in this manual.

Appendix B Describes how to set up the `svc.conf` file manually and by using the `svcsetup` script.

Appendix C Provides additional information on LAT.

Appendix D Provides additional information on writing automount maps.

Related Documents

For more information about networking and communications in the operating system, see the following manuals:

- *Network and Communications Overview* – Describes the DEC OSF/1 network environment, including network protocols and network services.
- *Command and Shell User's Guide* – Contains information on using network application programs.

- *Network Administration and Problem Solving* – Describes day-to-day management of the networking software on your DEC OSF/1 system, and how to recognize and solve common problems.

The printed version of the DEC OSF/1 documentation set is color coded to help specific audiences quickly find the books that meet their needs. (You can order the printed documentation from Digital.) This color coding is reinforced with the use of an icon on the spines of books. The following list describes this convention:

Audience	Icon	Color Code
General Users	G	Teal
System Administrators	S	Red
Network Administrators	N	Yellow
Programmers	P	Blue
Reference Page Users	R	Black

Some books in the documentation set help meet the needs of several audiences. For example, the information in some system books 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 books in the DEC OSF/1 documentation set.

Reader's Comments

Digital welcomes your comments on this or any other DEC OSF/1 manual. You can send your comments in the following ways:

- Internet electronic mail: `readers_comment@ravine.zk3.dec.com`
- Fax: 603-881-0120 Attn: USG Documentation, ZK03-3/Y32
- A completed Reader's Comments form (postage paid, if mailed in the United States). Two Reader's Comments forms are located at the back of each printed DEC OSF/1 manual.

If you have suggestions for improving particular sections or find any errors, please indicate the title, order number, and section numbers. Digital also welcomes general comments.

Conventions

The following conventions are used in this manual:

`%`
`$`
A percent sign represents the C shell system prompt. A dollar sign represents the system prompt for the Bourne and Korn 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).

Overview of Network Configuration 1

1.1 Overview of Tasks

Setting up your system to function fully in a network consists of setting up the following network components:

- Transmission Control Protocol/Internet Protocol (TCP/IP) network
- Local Area Transport (LAT)
- Berkeley Internet Name Domain (BIND) service
- Network Information Service (NIS)
- Network File System (NFS)
- UNIX-to-UNIX Copy Program (UUCP)
- Network Time Protocol (NTP)
- Mail
- Simple Network Management Protocol (SNMP) Agent

Note

> You must run the `netsetup` utility to configure your network before completing any other tasks. After the network is established, Digital recommends that you complete the tasks in the order that they appear in this guide.

Where setup utilities are available, you should use them to set up the components on your system.

1.2 Required Subsets

With the exception of LAT, UUCP, and mail, all the subsets needed to perform the tasks in this guide are mandatory subsets.

To configure LAT, UUCP, and mail, the following subsets are required:

- LAT — Local Area Transport (LAT) (OSFLAT)
- UUCP — UNIX to UNIX Copy Facility (OSFUUCP)

- Mail
  - RAND Corp. Mail Handler (OSFMH)
  - CDA Base Services (OSFCDABASE)
  - CDA Worksystem Base Services (OSFXCDA)
  - DECwindows Mail (OSFXMAIL)

1.3 Accessing the Setup Utilities

Most of the network software components provide setup utilities that automate the setup process. These utilities reside in the `/usr/sbin` directory.

You can invoke each utility by logging in as superuser and then entering the name of the utility. For example, to invoke the `netsetup` utility, you could do the following:

1. Log in as superuser.
2. Change the default directory to `/usr/sbin`.
3. Enter the `netsetup` command.

Alternatively, you can access the setup utilities from the Setup Menu. The Setup Menu also resides in the `/usr/sbin` directory. To access the setup utilities from the Setup Menu, perform the following steps:

1. Log in as superuser.
2. Select System Setup from the Applications Menu or enter the following command:

   ```
   # /usr/sbin/setup
   ```

   The Setup Menu appears as follows:

```
You can use this menu to set up your system and network.
Select the item you want to set up and answer the questions.

For more information on the items in the menu see the
"System Administration" and "Network Configuration."


         1) Internet Networking
         2) Local Area Transport (LAT)
         3) Berkeley Internet Name Domain Service (BIND)
         4) Network Information Service (NIS)
         5) Network File System (NFS)
         6) UNIX-to-UNIX Copy Program (UUCP)
         7) Network Time Protocol (NTP)
         8) Mail
         9) Simple Network Management Protocol (SNMP)
        10) Printers
        11) Streams
        12) License Management Facility (LMF)
        13) Verifier/Exerciser Tool (VET)
        14) Exit Setup Menu

Please enter your selection:
```

Note that if you do not have a particular subset installed, the menu item appears without an option number.

3. Enter the number for the component you want to configure.

Part 1: Suggested Setup Methods

Chapters 2 through 10 describe the suggested methods for setting up the following:

- Transmission Control Protocol/Internet Protocol (TCP/IP) network (Chapter 2)
- Local Area Transport (LAT) (Chapter 3)
- Berkeley Internet Name Domain (BIND) service (Chapter 4)
- Network Information Service (NIS) (Chapter 5)
- Network File System (NFS) (Chapter 6)
- UNIX-to-UNIX Copy Program (UUCP) (Chapter 7)
- Network Time Protocol (NTP) (Chapter 8)
- Mail (Chapter 9)
- Simple Network Management Protocol (SNMP) Agent (Chapter 10)

Setting Up the Network 2

You must run the `netsetup` utility to correctly configure your system to run on a network. During the network setup and configuration process the following files are created or modified:

- `/etc/hosts`
- `/etc/rc.config`

After the network interfaces are configured, you can choose to further customize the networking environment on the system by creating or modifying the following files:

- `/etc/hosts`
- `/etc/hosts.equiv`
- `/etc/networks`

Note

> Set up your network and make sure it is running before setting up the Berkeley Internet Name Domain (BIND) service, the Network Information Service (NIS), the Network File System (NFS), the UNIX-to-UNIX Copy Program (UUCP), the Network Time Protocol (NTP), or the Simple Network Management Protocol (SNMP) Agent.

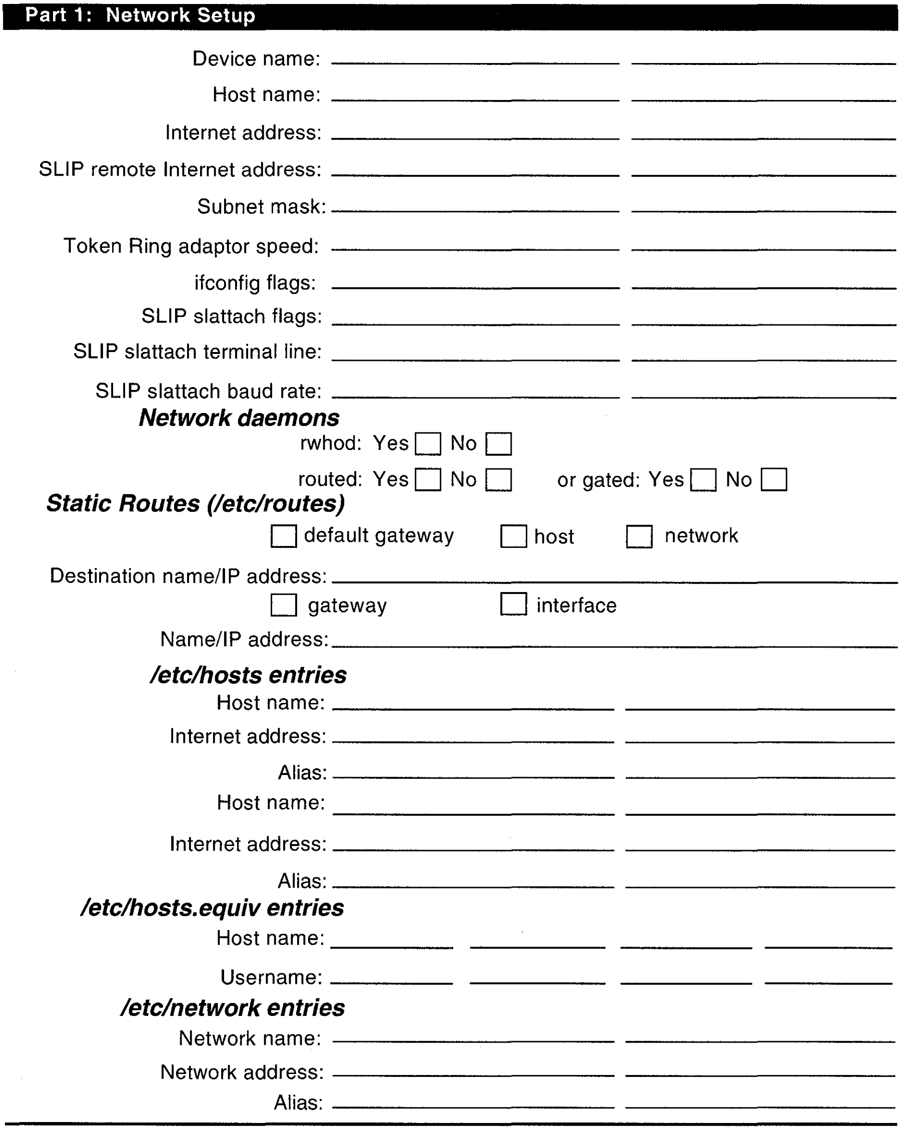
2.1 Gathering Information

Appendix A contains a worksheet that you can use to record the information that you need to complete the tasks in this book. Use Part 1 of the worksheet to record the information you gather as you work your way through this section. To obtain a copy of the worksheet, print the following PostScript file:

```
/usr/examples/network_configuration/worksheet.ps
```

Figure 2-1 shows Part 1 of the Configuration Worksheet.

Figure 2-1: Configuration Worksheet, Part 1



Part 1: Network Setup

Device name: ____________ ____________

Host name: ____________ ____________

Internet address: ____________ ____________

SLIP remote Internet address: ____________ ____________

Subnet mask: ____________ ____________

Token Ring adaptor speed: ____________ ____________

ifconfig flags: ____________ ____________

SLIP slattach flags: ____________ ____________

SLIP slattach terminal line: ____________ ____________

SLIP slattach baud rate: ____________ ____________

Network daemons

rwhod: Yes □ No □

routed: Yes □ No □ or gated: Yes □ No □

Static Routes (/etc/routes)

□ default gateway □ host □ network

Destination name/IP address: ________________________

□ gateway □ interface

Name/IP address: ________________________

/etc/hosts entries

Host name: ____________ ____________

Internet address: ____________ ____________

Alias: ____________ ____________

Host name: ____________ ____________

Internet address: ____________ ____________

Alias: ____________ ____________

/etc/hosts.equiv entries

Host name: ________ ________ ________ ________

Username: ________ ________ ________ ________

/etc/network entries

Network name: ____________ ____________

Network address: ____________ ____________

Alias: ____________ ____________

ZK-0759U-R

Gather the following information before you run the `netsetup` utility:

- The device names of the network interfaces

  The `netsetup` utility determines how many interfaces there are on your system and displays the device names. The utility then prompts you for the device name of the interface you want to configure.

- The host or interface name
- Your system's Internet Protocol (IP) address

  You should have obtained an IP address for your network from the Network Information Center (NIC). After you receive your network's address, you must assign a unique IP address and host name to each system on your network.

  To obtain an Internet address for your network, contact:

  Network Information Center
  Suite 200
  14200 Park Meadow Drive
  Chantilly, VA 22021

  Telephone numbers: (800) 365-3642 or (703) 802-4535

  FAX: (703) 802-8376

  E-mail: nic@nic.ddn.mil (For general information)
  hostmaster@nic.ddn.mil (For IP and domain registrations)

  In Europe, you can contact:

  RIPE Network Coordination Center
  Kruislaan 409
  NL-1098 SJ Amsterdam
  The Netherlands

  Telephone number: +31 20 592 5065

  FAX: +31 20 592 5090

  E-mail: ncc@ripe.net (For general information)
  Hostmaster@ripe.net (For IP and domain registrations)

  Note

  Digital recommends that you register your network with the NIC even if you do not intend to connect to the Internet network. Then, if you decide to connect to the Internet network later, you will not have to change all the host addresses on your network.

- You system's host name

  Each host on the network is assigned a unique name. A fully qualified hostname contains the host name and the domain name. The host name and each level of the domain name are separated by a period (.). Ask the network administrator for a unique host name.

  For more information, see the *Network and Communications Overview.*

- Your system's Serial Line Internet Protocol (SLIP) interface IP address.

  SLIP is used to run IP over serial lines. Each SLIP interface must have an IP address.

  For more information on SLIP, see Chapter 12, the *Network and Communications Overview*, and `slattach`(8).

- Your network's subnet mask

  Subnetworks allow the systems on a local area network (LAN) to be known by one address to the Internet network, while being known locally by a set of addresses. Subnetworks can represent logical groupings of hosts, or different physical networks. If your network uses subnetwork routing, each system on the network must have the same subnet mask defined.

  Use the following table to help identify your subnet mask. If you are not using subnetworks, the *n* is zero (0). Otherwise, the *n* is greater than zero and less than or equal to 255.

Class	**IP Address Range**	**Subnet Mask**
A	0.0.0.0 to 127.0.0.0	255.*n*.*n*.*n*
B	128.0.0.0 to 191.0.0.0	255.255.*n*.*n*
C	192.0.0.0 to 223.0.0.0	255.255.255.*n*

  If you are connecting your system to an existing network that is using subnetwork routing, ask the network administrator for the correct subnet mask.

  For more information on subnetwork routing see the *Network and Communications Overview.*

- Your system's token ring adapter speed

  If your system supports token ring, you must provide the speed of your system's token ring adapter. Only two speeds are supported: 4Mb/s and 16Mb/s. The default speed is 16Mb/s.

- Whether you want to add any flags to the `ifconfig` line

  The `ifconfig` command assigns an IP address to a network interface and configures network interface parameters. At boot time, the `ifconfig` command is run from the `/sbin/init.d/inet` file to define the network address of each interface. You can also use the `ifconfig` command at other times to redefine the address of an interface or to set other operating parameters.

  The default `ifconfig` line that the `netsetup` utility supplies is usually adequate; you can, however, add flags. For example, the standard broadcast address for a network is its IP address with the bits for the host portion of the address set to 1. If your network does not follow the standard, you should supply a new value for the –`broadcast` flag.

See the `ifconfig`(8) reference page for a description of the available flags.

- The SLIP options

  You must decide which SLIP options you want enabled or disabled. In making this decision, you must be aware of the options available on the remote system.

  The following options are available:

  - Internet Control Message Protocol (ICMP) traffic suppression — If you enable this option, ICMP traffic (such as that generated by the `ping` command) is not permitted to be sent over the SLIP connection. This frees line bandwidth for more critical traffic.
  - TCP header compression — If you enable this option, TCP headers are compressed before being sent over the SLIP connection, which allows for faster data transfers. The remote system must support this option to decompress the headers when they arrive at the remote end.
  - Auto enable of TCP header compression — If you enable this option, the local system compresses TCP headers when it detects that the remote system is compressing them. This option can be useful if you do not know whether the remote system is doing TCP header compression.

  Note

  > If the auto enable option is enabled on both systems, TCP header compression does not occur. One of the two systems must explicitly enable TCP header compression.

  For more information, see the `slattach`(8) reference page.

- The SLIP baud rate and terminal line specification

  You will need to know the baud rate of the modem (or null modem) used to connect the systems and the terminal line specification.

  The default baud rate is 9600 baud.

  The terminal line specification is the name of any valid terminal device in the `/dev` directory. This can be either the full path name (for example, `/dev/tty01`) or the name in the `/dev` directory (for example, `tty01`).

  For more information on the baud rate and the terminal line specification, see the `slattach`(8) reference page.

- The names and addresses of other hosts on the network

  You need to add the names and IP addresses of other hosts on your network to the /etc/hosts file.

  If your network is running a distributed database lookup service (BIND or NIS) you do not need to list each host on your network in your /etc/hosts file. However, even if your network is running a distributed database lookup service, it is a good idea to list four or five systems that you have designated in your /etc/hosts file as BIND or NIS servers.

- The names of trusted hosts

  Trusted hosts are listed in the /etc/hosts.equiv file. Systems listed in the /etc/hosts.equiv file are logically equivalent to, and therefore treated exactly the same as, the local system.

  Setting up an /etc/hosts.equiv file is optional, but, if you choose to have one on your system, you need to create it and add the names of any trusted hosts.

- Whether to run the rwhod daemon

  Running the rwhod daemon allows you to use the rwho and ruptime commands.

- Whether to run either the routed daemon or the gated daemon

  Running the routed daemon allows your system's internal routing tables for the Routing Information Protocol (RIP) to be updated automatically.

  Running the gated daemon allows your system's internal routing tables for different routing protocols to be updated automatically.

- Whether to add static routes

  The netsetup utility enables you to add static routes. A static route is a specific path from your system to another host or network that you define manually. A static route is not updated by network software.

  You configure static routes in the /etc/routes file, using the following information:

  - Whether you want to route to a default gateway, a host, or a network
  - The name or IP address of the destination to which you route

    If you are routing to a default gateway, the netsetup utility automatically sets the destination to the keyword default.

  - Whether you are routing through a gateway or an interface
  - The name or IP address of the gateway or interface

2.2 Running netsetup

You can invoke the `netsetup` utility by choosing the Internet Networking option from the Setup Menu or by issuing the following command:

```
# /usr/sbin/netsetup
```

When you invoke `netsetup`, the Main Menu is displayed.

To configure or delete a network interface, see Section 2.2.1.

To run network daemons or configure static routes, see Section 2.2.2.

To add or delete entries in the `/etc/hosts`, `/etc/hosts.equiv`, or `/etc/networks` file, see Section 2.2.3.

To display your current hardware and software configuration, see Section 2.2.4.

To exit `netsetup`, see Section 2.2.5.

2.2.1 Configuring and Deleting Network Interfaces

To configure or delete network interfaces, choose 1 from the Main Menu.

Note that the `netsetup` utility automatically creates any STREAMS devices in the `/dev/streams` directory that it finds on the system. The STREAMS devices that are configured are displayed along with their major and minor numbers.

The `netsetup` utility asks you a series of questions about the system. Default answers are provided in square brackets ([]). To use a default answer, press Return.

To configure network interfaces, see Section 2.2.1.1.

To delete network interfaces, see Section 2.2.1.2.

2.2.1.1 Configuring Network Interfaces

To configure interfaces, perform the following steps:

1. Indicate and confirm that you want to configure network interfaces.

   Enter `c` at the prompt that asks whether you want to configure or delete interfaces.

   The utility prompts you for confirmation. To configure network interfaces, press Return. Otherwise, enter `n`.

2. Enter and confirm the device name of the interface you want to configure.

   If all the network interfaces are configured, the utility asks if you want to reconfigure an interface. If you do not want to reconfigure an interface, press Return. Otherwise, enter `y`.

The `netsetup` utility lists the network adapters that are on the system and asks you to enter the device name of the interface you want to configure. To configure the default device, press Return.

The utility displays the device name of the interface you selected and asks for confirmation.

If the interface you chose is already configured, the utility asks if you want to reconfigure this interface. If you do not want to reconfigure this interface, press Return. Otherwise, enter `y`.

3. Enter and confirm the host name or interface name.

   If this is the first or only network interface you are configuring and there is a default host name, the utility displays the default. To take the the default host name, press Return. To specify a different host name, enter the name.

   If there is no default host name, the `netsetup` utility prompts you for one.

   The `netsetup` utility displays the host name and asks you to confirm that the name is correct.

   If this is not the first network interface you are configuring, the utility asks you to enter a name for the interface you want to configure. If you do not want to name the interface, press Return. The utility asks you to confirm that you do not want to name the interface.

   The `netsetup` utility displays the name and asks you to confirm that it is correct.

4. Enter and confirm the Internet Protocol (IP) address.

   If there is a default IP address, the utility displays the default. To take the default address, press Return. To specify a different address, enter the address.

   If there is no default IP address, `netsetup` prompts you for one.

   The utility displays the IP address and asks you to confirm that it is correct. If it is, press Return. If not, enter `n`. The utility then prompts you again for the IP address.

5. For Serial Line Internet Protocol (SLIP) interfaces, enter and confirm the IP address of the remote SLIP interface.

   If there is a default IP address, the utility displays the default. To take the default address, press Return. To specify a different address, enter the address.

   If there is no default IP address, `netsetup` prompts you for one.

   The utility displays the IP address and asks you to confirm that it is correct.

6. Enter and confirm the subnet mask.

   The `netsetup` utility asks you to enter the subnet mask. To accept the default, press Return. Otherwise, enter the subnet mask.

   The utility displays the subnet mask and asks you to confirm that it is correct. If it is, press Return. If not, enter `n`. The utility then prompts you again for the subnet mask.

7. For token ring interfaces, enter and confirm the speed of the adapter.

   The utility asks you to enter the speed. To take the default speed, press Return. To specify a different speed, enter the speed.

8. Indicate whether you want to add additional `ifconfig` flags.

   The `netsetup` utility asks whether you want to add additional `ifconfig` flags. If you do not want additional flags, press Return. To add additional flags, enter `y`. Enter the flags when prompted, separating each flag with a space.

   The utility displays the `ifconfig` flags and asks you to confirm that they are correct.

9. Confirm that the interface configuration is correct.

   The `netsetup` utility displays the configuration parameters and asks you to confirm that they are correct.

10. For SLIP interfaces, enter and confirm the `slattach` command flags.

    If you do not want `slattach` flags, press Return. Otherwise, enter the flags and press Return.

    For more information on `slattach` command flags, see `slattach`(8).

11. For SLIP interfaces, enter and confirm the `slattach` command terminal name.

12. For SLIP interfaces, enter and confirm the `slattach` command baud rate.

    The utility displays the default baud rate of 9600 baud and prompts you to confirm that this is correct.

13. Confirm that the `slattach` command is correct.

    If the `slattach` command is correct as displayed, press Return. If not, enter `n`. The utility prompts you again for the `slattach` command parameters.

    For more information, see `slattach`(8).

To return to the Main Menu, press Return at the prompt that asks whether you want to configure or delete interfaces.

2.2.1.2 Deleting Network Interfaces

To delete interfaces, perform the following steps:

1. Enter d at the prompt that asks whether you want to configure or delete interfaces.
2. Press Return to confirm that you want to delete an interface.
3. If there is only one interface configured, the utility informs you of this and asks if you want to continue. To delete the interface, enter y.

   If you do not want to delete the interface, press Return.
4. Indicate the interface you want to delete.

   The utility displays a list of configured interfaces and provides a default interface for you to delete. To delete the default interface, press Return. To delete a different interface, enter the interface name.
5. Indicate whether you want to delete another interface.

   To delete another interface, enter y, and go to step 3.

   If you do not want to delete another interface, press Return.

To return to the Main Menu, press Return at the prompt that asks whether you want to configure or delete interfaces.

2.2.2 Enabling or Disabling Network Daemons and Adding Static Routes

The `netsetup` utility allows you to do the following:

- Enable or disable the `rwhod` daemon.
- Enable or disable either the `routed` daemon or the `gated` daemon.
- Add static routes.

To enable the `gated` daemon, set up the `/etc/gated.conf` file in the format specified in `gated.conf`(4). The `netsetup` utility creates a default `/etc/gated.conf` file, if it does not exist.

To perform these tasks, choose 2 from the Main Menu and perform the following steps:

1. Indicate whether you want to run the `rwhod` daemon.

   If you do not want to run the `rwhod` daemon, press Return.

   To run the `rwhod` daemon enter y.
2. If your system has more than one network interface, indicate whether you are setting up your system to be an IP router.

   If you want your system to be an IP router, enter y.

If you do not want your system to be a router, press Return.

Note

> The `netsetup` utility sets the `ROUTER` variable in `/etc/rc.config`, and sets the `ipforwarding` and `ipgateway` variables in the kernel.

3. Indicate whether you want to run the `gated` daemon or the `routed` daemon.

   You can not run both the `gated` daemon and the `routed` daemon.

   If you do not want to run the `gated` daemon or the `routed` daemon, press Return at the prompt that asks if you want to run the `gated` daemon or the `routed` daemon.

   To run the `gated` daemon, enter `g` at the prompt that asks if you want to run the `gated` daemon or the `routed` daemon. When prompted, enter the `gated` daemon flags that you want. Separate each flag with a space.

   To run the `routed` daemon, enter `r` at the prompt. When prompted, enter the `routed` daemon flags that you want. Separate each flag with a space.

4. Indicate whether you want to add static routes.

   If you do not want to add any more static routes, press Return.

   To add static routes, enter `y`.

   To add static routes, configure each route in `/etc/routes` by performing the following steps:

   a. Indicate whether you want to route to a default gateway (enter `d`), a host (enter `h`), or a network (enter `n`).

   b. Specify the destination name or IP address.

      If you are routing to a default gateway, the destination is automatically set to the keyword `default` by `netsetup`.

   c. Indicate whether you are routing through a gateway or an interface.

   d. Specify the name or IP address of the gateway or interface.

   e. Confirm that the static route is correct.

   If you do not want to add more static routes, press Return at the prompt that asks if you want to add static routes.

When you have finished adding static routes, the utility returns you to the Main Menu.

2.2.3 Adding or Deleting Host Information

The `netsetup` utility enables you to add or delete host information from the following files:

- `/etc/hosts`
- `/etc/hosts.equiv`
- `/etc/networks`

To perform any of these functions, choose 3 from the Main Menu. The `netsetup` utility presents you with the Host Information Menu.

To add or delete entries in the `/etc/hosts` file, see Section 2.2.3.1.

To add or delete entries in the `/etc/hosts.equiv` file see Section 2.2.3.2.

To add or delete entries in the `/etc/networks` file, see Section 2.2.3.3.

To return to the Main Menu, choose 4.

2.2.3.1 Adding and Deleting /etc/hosts Entries

To add or delete host information from the `/etc/hosts` file, choose 1 from the Host Information Menu.

Section 2.2.3.1.1 explains how to add information to the `/etc/hosts` file.

Section 2.2.3.1.2 explains how to delete information from the `/etc/hosts` file.

2.2.3.1.1 Adding Entries to /etc/hosts – To add an entry to the `/etc/hosts` file, perform the following steps:

1. Enter `a` at the prompt that asks if you want to add or delete hosts in the `/etc/hosts` file.
2. Enter the name of the host to add to the `/etc/hosts` file.
3. Enter any aliases for the host. Separate each alias with a space.
4. Enter the Internet address for the host in dot notation.
5. Confirm that the information is correct.

After you add the host, you can add another host or, if you are finished adding hosts, press Return. The utility prompts you to indicate whether you want to add or delete another host. If you are finished adding and deleting host information, press Return to return to the Host Information Menu.

2.2.3.1.2 Deleting Entries from /etc/hosts – To delete an entry from the `/etc/hosts` file, perform the following steps:

1. Enter `d` at the prompt that asks if you want to add or delete hosts in the `/etc/hosts` file.
2. Enter the name of the host to delete from the `/etc/hosts` file.

After you delete the host, you can delete another host or, if you are finished deleting hosts, press Return. The utility prompts you to indicate whether you want to add or delete another host. If you are finished adding and deleting host information, press Return to return to the Host Information Menu.

2.2.3.2 Adding and Deleting /etc/hosts.equiv Entries

To add or delete host information from the `/etc/hosts.equiv` file, choose 2 from the Host Information Menu.

Section 2.2.3.2.1 explains how to add information to the `/etc/hosts.equiv` file.

Section 2.2.3.2.2 explains how to delete information from the `/etc/hosts.equiv` file.

2.2.3.2.1 Adding Entries to /etc/hosts.equiv – To add an entry to the `/etc/hosts.equiv` file, perform the following steps:

1. Enter `a` at the prompt that asks if you want to add or delete hosts in the `/etc/hosts.equiv` file.
2. Enter the name of the host to add to the `/etc/hosts.equiv` file.

   If the host is not on the network, you cannot add the host.
3. Enter the login name of a trusted user.

   If you do not want to specify a trusted user, press Return when prompted for this information.

After you add the host and trusted user, you can add another host or, if you are finished adding hosts, press Return. The utility prompts you to indicate whether you want to add or delete another host. If you are finished adding and deleting host information, press Return to return to the Host Information Menu.

2.2.3.2.2 Deleting Entries from /etc/hosts.equiv – To delete an entry from `/etc/hosts.equiv`, perform the following steps:

1. Enter `d` at the prompt that asks if you want to add or delete hosts in the `/etc/hosts.equiv` file.

2. Enter the name of the host to delete from the `/etc/hosts.equiv` file.

After you delete the host, you can delete another host or, if you are finished deleting hosts, press Return. The utility prompts you to indicate whether you want to add or delete another host. If you are finished adding and deleting host information, press Return to return to the Host Information Menu.

2.2.3.3 Adding and Deleting /etc/networks Entries

To add or delete networks in the `/etc/networks` file, choose 3 from the Host Information Menu.

Section 2.2.3.3.1 explains how to add information to the `/etc/networks` file.

Section 2.2.3.3.2 explains how to delete information from the `/etc/networks` file.

2.2.3.3.1 Adding Entries to /etc/networks – To add an entry to the `/etc/networks` file, perform the following steps:

1. Enter `a` at the prompt that asks if you want to add or delete networks in the `/etc/networks` file.
2. Enter the name of the network to add to the `/etc/networks` file.
3. Enter any aliases for the network. Separate the aliases with a space.
4. Enter the network number for the network.
5. Confirm that the information is correct.

After you add the network, you can add another network or, if you are finished adding networks, press Return. The utility prompts you to indicate whether you want to add or delete another network. If you are finished adding and deleting network information, press Return to return to the Host Information Menu.

2.2.3.3.2 Deleting Entries from /etc/networks – To delete an entry from the `/etc/networks` file, perform the following steps:

1. Enter `d` at the prompt that asks if you want to add or delete hosts in the `/etc/networks` file.
2. Enter the name of the network to delete from the `/etc/networks` file.

After you delete the network, you can delete another network or, if you are finished deleting networks, press Return. The utility prompts you to indicate whether you want to add or delete another network. If you are finished adding and deleting network information, press Return to return to the Host Information Menu.

2.2.4 Displaying Network Configuration Information

To display network configuration information, choose 4 from the Main Menu. The `netsetup` utility displays a list of network adapters on the system along with the following information from the `/etc/rc.config` file:

- HOSTNAME – Specifies the host name of the system.
- NUM_NETCONFIG – Specifies the number of network interfaces currently configured.
- MAX_NETDEVS – Indicates the maximum number of network devices that can be configured. The maximum is 16.
- NETDEV_*n* – Specifies the network device name for the network interfaces currently configured. The value of *n* is 0 to 1 less than the value of `MAX_NETDEVS`.
- IFCONFIG_*n* – Specifies the `ifconfig` parameters for the network interfaces currently configured. The value of *n* is 0 to 1 less than the value of `MAX_NETDEVS`.
- RWHOD – Indicates whether the `rwhod` daemon is enabled. If the daemon is enabled, `yes` is displayed. If the daemon is disabled, either a blank or `no` is displayed.
- ROUTED – Indicates whether the `routed` daemon is enabled. If the daemon is enabled, `yes` is displayed. If the daemon is disabled, either a blank or `no` is displayed.
- ROUTED_FLAGS – Displays any `routed` daemon flags that are configured. (See the `routed`(8) reference page for more information.)
- GATED – Indicates whether the `gated` daemon is enabled. If the daemon is enabled, `yes` is displayed. If the daemon is disabled, either a blank or `no` is displayed.
- GATED_FLAGS – Displays any `gated` daemon flags that are configured. (See the `gated`(8) reference page for more information.)
- ROUTER – Indicates whether the system is set up to be an IP router.
- SLIPTTY_*n* – Specifies the `slattach` command parameters for SLIP network interfaces currently configured. The value of *n* is 0 to 1 less than the value of MAX_NETDEVS. (See `slattach`(8) for more information.)

2.2.5 Exiting netsetup and Starting Network Services

To exit the `netsetup` utility, choose 5 from the Main Menu.

For your changes to take effect, you must restart the network. To do this automatically when you exit netsetup, enter y when netsetup asks if you want to automatically restart the network services on this system. Alternatively, you can start the network services later by issuing the following command:

```
# /usr/sbin/rcinet restart
```

2.3 More Information

For more information, see netstat(1), gated(8), ifconfig(8), slattach(8), routes(4), and routed(8).

For information about IP addresses, subnetworks, network classes, and routing, see the *Network and Communications Overview*.

For information about network management, see the *Network Administration and Problem Solving* manual.

Setting Up the Local Area Transport 3

The Local Area Transport (LAT) is a protocol that provides an efficient means of logically connecting terminal servers to one or more nodes on the same local area network (LAN). LAT software has the features required for a host to function as a service node, so requests for connections can be made by server users. LAT software also permits host applications to initiate connections to server ports, designated as applications ports, to access remote devices.

This chapter provides information on the followng:

- Gathering information required to configure LAT
- Configuring LAT support in your kernel
- Using the `latsetup` utility to setup LAT on your system
- Customizing your LAT configuration
- Controlling access in a LAT network

During the LAT set up and configuration process the following files are created or modified:

- `/etc/inittab`

  The `/etc/inittab` file controls the initialization process.

- `/etc/rc.config`

  The `/etc/rc.config` file specifies the system configuration file.

- `/dev/tty`*`WX`*

  The `/dev/tty`*`WX`* file specifies the LAT terminal devices. (The value of *W* is a number from 0 to 9 and *X* is an alphanumeric from 0 to 9, a lowercase a to z, or an uppercase A to Z.)

- `/dev/streams/kinfo`

  The `/dev/streams/kinfo` file specifies the STREAMS pseudodevice.

The `latsetup` utility also automatically creates STREAMS devices in the `/dev/streams` directory by running `strsetup`. LAT requires the `/dev/streams/kinfo` STREAMS device.

Note

You must install the LAT subset and configure LAT support in the running kernel prior to setting up LAT. (See Section 3.2.)

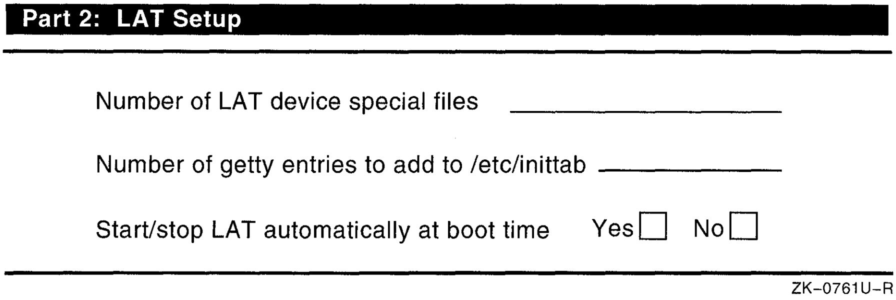
3.1 Gathering Information

Appendix A contains a worksheet that you can use to record the information that you need to complete the tasks in this book. Use Part 2 of the worksheet to record the information you gather as you work your way through this section. To obtain a copy of this worksheet, print the following PostScript file:

```
/usr/examples/network_configuration/worksheet.ps
```

Figure 3-1 shows Part 2 of the Configuration Worksheet.

Figure 3-1: Configuration Worksheet, Part 2



Part 2: LAT Setup

Number of LAT device special files ____________________

Number of getty entries to add to /etc/inittab ______________

Start/stop LAT automatically at boot time Yes □ No □

ZK-0761U-R

Gather the following information before running the `latsetup` utility:

- Determine the number of LAT device special files you need.

  The LAT device special files can be used for incoming connections and for host-initiated connections to remote devices. You must determine the total number of LAT device special files to be created to be used for both incoming connections and host-inititated connections.

- Determine the number of `getty` entries to add to the `/etc/inittab` file.

  The `getty` entries in the `/etc/inittab` file specify the LAT terminal lines that are used for incoming connections.

  You must determine the number of LAT terminal lines needed for the users and processes (such as the LAT/Telnet gateway) on your system.

  As your user community grows and each user wants to run multiple sessions on one or more timesharing machines in your environment, your system might run out of available LAT device special files. When a user tries to connect using the LAT protocol to a timesharing machine that does not have enough LAT `getty` lines spawned, the timesharing system returns the following error message at the server:

```
Insufficient resources
```

- Determine if you want LAT to be started or stopped automatically at boot time.

  When you enable LAT automatic startup and shutdown, the `/sbin/init.d/lat` startup and shutdown script automatically starts LAT upon reaching run level 3 and automatically stops LAT when exiting run level 3.

3.2 Configuring Your Kernel

To configure LAT support in your kernel, log in as superuser and complete the following steps:

1. Make certain that the configuration file contains the LAT option.

   When you install the LAT subset and execute the `doconfig` command to build a kernel without specifying a configuration file, `doconfig` automatically puts the LAT option (`options LAT`) in the configuration file.

   If you specify a configuration file when you run `doconfig`, you must ensure that the configuration file contains the LAT option. The default configuration file might already contain the LAT option (`options LAT`) in a line that has been commented out. In that case, remove the comment symbol (#) from the beginning of the line.

   If your configuration file does not contain the LAT option, you must add it before you build the kernel. The syntax for the LAT option is as follows:

   options LAT

   The default configuration file to edit is `/sys/conf/`*HOSTNAME*. (*HOSTNAME* is the name of your host processor, in uppercase letters.)

2. After updating the configuration file, build a new kernel. For more information on how to build the kernel, see the *System Administration* manual.

3. Reboot your system with the new kernel by issuing the following

command:

```
# shutdown -r now
```

This command immediately shuts down and automatically reboots the system.

3.3 Running latsetup

You use the `latsetup` program to administer LAT on your system. To use `latsetup`, LAT must be built into the running kernel, your system must be at run level 3, and you must be logged in as superuser.

The `latsetup` utility allows you to do the following:

- Create LAT device special files
- Add or remove `getty` entries to or from the `/etc/inittab` file
- Execute `init q`
- Start or stop the LAT driver
- Enable or disable LAT automatic startup and shutdown

To invoke the `latsetup` utility choose the Local Area Transport (LAT) option from the Setup Menu or enter the following command:

```
# /usr/sbin/latsetup
```

If your terminal does not support curses, you must specify the `-nocurses` flag. This flag allows you to run `latsetup` in noncurses and nonmenu-driven mode.

Note

> Running multiple `latsetup` processes concurrently on the same machine can cause erroneous information to be presented to the `latsetup` user and can corrupt the `/etc/inittab` file.

For more information, see the `latsetup`(8) reference page.

3.4 Customizing LAT

This section provides the following information on how to customize LAT on your system:

- Performing general customization
- Setting up printers
- Setting up host-initiated connections

- Setting up the LAT/Telnet gateway
- Creating your own service

3.4.1 Customizing Your System

After you run `latsetup`, you can customize your system's LAT environment by modifying either or both the `/etc/inittab` file and the `/sbin/init.d/lat` startup and shutdown script.

You can use the LAT utility `latcp` to modify to the `/sbin/init.d/lat` startup and shutdown script. When LAT automatic startup and shutdown is enabled, the `/sbin/init.d/lat` script starts LAT upon reaching run level 3. LAT automatic startup and shutdown can be enabled or disabled with the `latsetup` utility. By default, this script contains the `latcp -r` and `latcp -s` commands. If the following parameters have not been set, they have the following default values when the `latcp -r` command is executed:

- Node name – *HOSTNAME*
- Multicast timer – 60 seconds
- Selected interface names – All the network Ethernet adapters on the system
- Service name – *HOSTNAME*
- Service ID – DEC OSF/1 Version X.X LAT SERVICE
- Rating – Dynamic
- Groups – 0

HOSTNAME is the name of your system in uppercase letters.

If your system is configured with multiple network adapters, you can specify that the LAT protocol run over the multiple adapters, provided the adapters are connected to different logical networks. If more than one network adapter is connected to a single logical network, you should use the `latcp` command to specify that the LAT protocol run over only one adapter. (See the `latcp`(8) reference page for more information.) You can determine the adapters defined on your system by using the `netsetup`(8) command.

You can modify the `/sbin/init.d/lat` script to include `latcp` commands to customize your LAT system. For example, you can define a particular node name or add service names. For more information, see `latcp`(8). Example 3-1 provides a portion of the script with additional lines for customization.

Example 3-1: Portion of /sbin/init.d/lat with Customization

```
if [ "$9" = "S" ]
then  if [ "$8" = "0" ]
      then # This code is only executed the *first* time the
           # system enters run level 3.
   •
   •
   •
           /usr/sbin/latcp -r   [1]
           /usr/sbin/latcp -n testnode   [2]
           /usr/sbin/latcp -A -a lattelnet14 -i "LAT/telnet" -o   [3]
           /usr/sbin/latcp -A -a testservice   [4]
           /usr/sbin/latcp -g 0,21,52 -a testservice   [5]
        fi

        # The following line(s) get executed everytime
        # '/sbin/init.d/lat start' is executed.
       /usr/sbin/latcp -s
fi ;;
```

In Example 3-1, the following commands are added to the `/sbin/init.d/lat` file for customization:

$\boxed{1}$ `/usr/sbin/latcp -r` sets up the following default LAT parameters: node name, multicast timer, network interfaces, and the default service with the default service id, rating, and groups.

$\boxed{2}$ `/usr/sbin/latcp -n testnode` changes the LAT node name.

$\boxed{3}$ `/usr/sbin/latcp -A -a lattelnet14 -i "LAT/telnet" -o` adds an optional service that can be used for LAT/Telnet connections. (See Section 3.4.4 for more information on the LAT/Telnet gateway.)

$\boxed{4}$ `/usr/sbin/latcp -A -a testservice` adds a service.

$\boxed{5}$ `/usr/sbin/latcp -g 0,21,52 -a testservice` adds groups 0, 21, and 52 to the service `testservice`.

If any `latcp` commands that require a service name (such as the `latcp -g` command) are added to the `/sbin/init.d/lat` script, the service must be added before the command is executed.

You can modify the `/etc/inittab` file to use a program other than `getty`. For example, you can add the following entry to `/etc/inittab` to set up `tty14` to use the user-defined program `myownprogram`:

```
lat14:3:respawn:/usr/sbin/myownprogram  tty14
```

For more information on using user-defined programs with LAT, see Section 3.4.5.

You can also modify the `/etc/inittab` file to add LAT devices created manually after the initial setup and device creation by adding an entry, similar to the following:

```
lat07:3:respawn:/usr/sbin/getty  tty07 console vt100
```

For more information, see the `inittab`(4) reference page.

3.4.2 Setting Up Printers

This section provides the following information on how to set up a printer to print through LAT, using host-initiated connections:

- Setting up the remote printer on a terminal server
- Testing the port configuration
- Setting up the DEC OSF/1 service node (local LAT host) for the printer
- Setting up the print spooler on the service node
- Testing the printer

Note

> The examples in this section use the DECserver 700 server. Please refer to the documentation supplied for your terminal server.

3.4.2.1 Setting Up the Remote Printer on a Terminal Server

Before you set up the printer, you must install it on a serial interface on a terminal server.

Using the appropriate terminal server commands, you must set up the server to allow access to the attached remote printer through host-initiated requests from the DEC OSF/1 service node. (Service node refers to the local DEC OSF/1 LAT host.) For information on how to do this, refer to your terminal server documentation.

After you set up the printer and the terminal server, you will need the following information:

- The name of the terminal server to which the printer is attached
- Either or both of the following:
  - The name of the port to which the printer is attached
  - The name of the service assigned for the remote printer

You must also match the hardware settings of the printer and the terminal server. To do this, you need to determine your printer's character size, flow control, parity, and speed. Refer to your printer documentation for this information.

After you determine your printer's characteristics, compare them to the terminal server's port settings. Be sure the settings correspond. You can see the settings on the terminal server console by using a command similar to the following:

```
Local> SHOW PORT 7 CHARACTERISTICS
```

This command shows the characteristics for port 7. At a minimum, the terminal server should have settings for the port similar to the following:

Character Size:	Printer's character size
Flow Control:	XON (or –CTS/RTS, for some printers)
Speed:	Printer's speed
Access:	Remote
Autobaud:	Disabled
Autoconnect:	Disabled

To permanently define a terminal server's port settings, use the DEFINE command. For example:

```
Local> DEFINE PORT 7 SPEED 9600
```

After you define the settings for the port, log out of that port to initialize the new settings. For example:

```
Local> LOGOUT PORT 7
```

3.4.2.2 Testing the Port Configuration

You need to test the port configuration to verify that the printer characteristics match in the printer and in the terminal server port.

Verify the port configuration by using the TEST PORT command on the terminal server. For example, if the configuration is correct, the following command run on a DECserver 700 prints a test pattern of characters on a printer attached to port 7:

```
Local> TEST PORT 7
```

The printer prints 24 lines of test data unless you press the Break key at the terminal server console. If data does not print or if it appears to be incorrect, the port or the printer is incorrectly set, or there is a hardware problem.

3.4.2.3 Setting Up a Service Node for the Printer

To set up the DEC OSF/1 service node (local LAT host) for the printer, you need the name of the terminal server and either the name of the port or the name of the service for the printer that was set up in Section 3.4.2.1.

Using latcp, map an unused application port (a tty created by latsetup) with the remote port, remote service, or both on the terminal server.

For example, the following command maps the local applications port (tty24) for the local server (LOCSER) to the remote printer port (port06).

```
# latcp -A -p tty24 -H LOCSER -R port06
```

The following command does the same thing, however, the remote printer service name is used:

```
# latcp -A -p tty24 -H LOCSER -V REMprinter06
```

Note

> The application port you specify cannot be used to spawn a getty operation in the /etc/inittab file.

For more information, see latcp(8).

3.4.2.4 Setting Up the Print Spooler on the Service Node

Use lprsetup to set up the print spooler for the remote printer. The following printcap symbols must be set for the DEC OSF/1 service node (local LAT host) to access the remote printer through host-initiated connections:

- ct — Connection type
- lp — Device name to open for output

You must specify LAT for the ct symbol. The lp symbol must be set to the LAT application port defined in Section 3.4.2.3. In the example in Section 3.4.2.3, lp is set to /dev/tty24.

Following is an example of an /etc/printcap entry for a LAT printer:

```
lp25|lp0:\
        :af=/usr/adm/lpacct:\
        :ct=LAT:\
        :lf=/usr/adm/lperr:\
        :lp=/dev/tty24:\
        :mx#0:\
        :of=/usr/lbin/lpf:\
        :sd=/usr/spool/lpd:
```

3.4.2.5 Testing the Printer

After you set up the printer, print a file to be sure everything works properly. For example, if the printer name is lp25 and test is a text file, you can test the printer by issuing the following command:

```
# lpr -Plp25 test
```

If the printer does not work, check to make sure all the settings are correct. If the `printcap` entry has an `lf` entry defined, you can check the corresponding file for information on errors that could have occurred.

3.4.3 Setting Up Host-Initiated Connections

This section describes how you set up a DEC OSF/1 system for host-initiated connections to any bit-serial, asynchronous device connected to a terminal server. Examples of such devices are terminals, modems, communications ports on other host computer systems, and printers. The printer connections discussed in Section 3.4.2 are one instance of a host-initiated connection.

This feature allows you to associate a named port on a named terminal server with a specific terminal device special file. As a result, users can develop applications that connect to the port through LAT. The type of device the target shows is transparent to the LAT protocol.

Example C-3 provides an example of an application that can be used with host-initiated connections.

3.4.3.1 Setting Up the System for Host-Initiated Connections

To define the connection between the host terminal and the terminal server port service, you run the LAT control program, `latcp`, using the `-A` option. In the command, you specify the applications port (`tty42`), the terminal server name (`T1301A`), and either the terminal port name (`PORT_6`) or the service name (`printer`), in that order. For example:

```
# /usr/sbin/latcp -A -p tty42 -HT1301A -R PORT_6
```

The protection bits, the owner, and the group of the terminal should be set appropriately for the intended use of the connection. For example, terminals are normally owned by root and are readable only by their owner. If you intend to let ordinary users open and read the terminal, you should make the terminal world readable.

Next, you must set up the server port characteristics to match the characteristics of the device connected to the port and to allow host-initiated connections. See your device and terminal server documentation.

3.4.3.2 The Program Interface

Applications developed to employ host-initiated connections are much like applications for any terminal device. However, there are some programming considerations:

- The programs communicate with the LAT driver through the device special file. When the host program issues an open call to the terminal, the LAT driver attempts to establish a connection to the target port or service on the target server. The driver reports success and failure codes in the variable `errno`.
- When the open call is successful, the user program issues `read` and `write` system calls to handle data transfers, and normal `ioctl` processing for the device control information.
- A close system call on the device terminates the LAT connection.

3.4.4 Setting Up the LAT/Telnet Gateway

This section describes how to set up and use the LAT/Telnet gateway service. By employing this service, a user with a LAT terminal server can connect directly to remote hosts through the Telnet protocol. The user does not have to log in to a local DEC OSF/1 system first. Optionally, you can use the `rlogin` command to connect directly to remote hosts.

For example, a user traveling on business could use a terminal on a LAN to connect through Telnet to her home system and account, even though she does not have an account on any system in the LAN.

To setup the LAT/Telnet gateway, perform the following steps:

1. Define the LAT/Telnet service.

   Use the `latcp` command to define the LAT/Telnet service. For example:

   ```
   /usr/sbin/latcp -A -a lattelnet -i "LAT/telnet gateway" -o
   ```

   The `-o` flag specifies that this is an optional service. Optional services are unlike default services in that they cannot be used to connect to the DEC OSF/1 local LAT host through `getty` lines spawned in the `/etc/inittab` file.

2. Edit the `/etc/inittab` file.

   Select the LAT terminals to dedicate to the gateway, for example `tty20`, `tty21`, and `tty22`. The number of terminals selected determines the maximum number of simultaneous LAT/Telnet gateway sessions the system can deliver.

   Edit the system's `/etc/inittab` file to include entries to spawn `lattelnet` on the selected devices. For example:

   ```
   lat20:3:respawn:/usr/sbin/lattelnet  tty20  lattelnet
   lat21:3:respawn:/usr/sbin/lattelnet  tty21  lattelnet
   lat22:3:respawn:/usr/sbin/lattelnet  tty22  lattelnet
   ```

   In the previous example, the last entry in each line (`lattelnet`) is the name of the optional service defined in step 1.

If you use the rlogin command (instead of Telnet) the /etc/inittab entry must specify /usr/bin/rlogin as the third argument to the lattelnet program. For example:

```
lat20:3:respawn:/usr/sbin/lattelnet tty20 lattelnet /usr/bin/rlogin
```

3. Start the gateway.

   Use the init q command to effect the changes to start up the gateway, as follows:

   ```
   # init q
   ```

   Use the ps(1) command to verify that the lattelnet process has started.

   The lattelnet program uses the syslog(3) function to log messages to the /var/adm/syslog.dated/daemon.log file. Check this file to verify that no error messages have been generated.

4. Connect to the gateway.

   To use the gateway from the LAT terminal server, enter the CONNECT command. For example, to connect to a remote node named REMOTE by using a local node named LOCAL as a gateway, enter:

   ```
   Local> CONNECT LATTELNET NODE LOCAL DEST REMOTE
   ```

   Alternatively (for LATTELNET), enter the service name LATTELNET and wait to be prompted for the remote node desired. The following example represents what occurs when a user on a terminal server connects to the service LATTELNET and waits for a login prompt from remote node MYTRIX:

   ```
   Local> CONNECT LATTELNET
   LAT to TELNET gateway on printf
   telnet> OPEN MYTRIX
   Trying...
   Connected to mytrix.
   Escape character is '^]'.
   mytrix login:
   ```

   If you use the rlogin command you must specify the NODE and DEST.

3.4.5 Creating Your Own Optional Service

The latcp command allows service nodes to offer multiple services. You can offer two different classes of services, as follows:

- A default service, which is used for normal interactive connections to DEC OSF/1 local LAT hosts through getty lines spawned in the /etc/inittab file.
- Optional services, which are used with specialized applications, written especially for LAT. One such service, a component of the operating

system software, is the LAT/Telnet gateway described in Section 3.4.4. By employing this service, a user on a LAT terminal server can connect directly to a remote node through Telnet protocols without having to log in first to a DEC OSF/1 system.

You can also write your own specialized applications and advertise them to terminal servers.

3.4.5.1 Programming the Service

Programming for a service can be as simple or as complex as the service you have designed. Examples of specialized applications that can be used with optional services are shown in Section C.1 and Section C.2.

3.4.5.2 Setting Up the Service

The following steps you take to set up a service are similar to those you take to set up the LAT/Telnet gateway discussed in Section 3.4.4:

1. Use the `latcp` command to set up the service. For example:

   ```
   # /usr/sbin/latcp -A -a showdate -o
   ```

2. Select the LAT terminals to be dedicated to the service.
3. Edit the system's `/etc/inittab` file to replace `getty` with the name of your service.
4. Use the `init q` command to make the changes take effect.

To use the service at a LAT terminal, issue the `CONNECT` command. For example:

```
Local> CONNECT SHOWDATE
```

3.5 Controlling Access in a LAT Network

Because LAT networks are local in nature, you have a high degree of control over the LAT environment and who has physical access to LAT devices. In addition to controlling physical access, two are features available that increase your control of LAT access:

- LAT terminal server login password

  You can require users to enter a password to gain access to terminal servers. (Refer to the documentation supplied with your terminal server.)

- LAT groups

  You can establish LAT groups and then restrict host communication to particular groups by designating those groups on a LAT host (by issuing a `latcp -g` command) on the terminal server. (Refer to the documentation supplied with your terminal server.)

Groups are used to partition the LAT network into logical subdivisions. Groups are set up by the network manager, system manager, and server managers. Groups are used to restrict message traffic between servers and service nodes. For a connection to be established, the terminal server requesting a connection to a LAT service node must share at least one group with that node. When messages are received by a terminal server from service nodes that are not in any group enabled on the server, these messages are ignored. Groups help manage the size of the servers' LAT databases by limiting the number of service nodes for which the server keeps information. Groups are not intended as a security mechanism.

For more information on LAT configuration, refer to the `latcp`(8) reference page.

Setting Up the Berkeley Internet Name Domain Service 4

The Berkeley Internet Name Domain (BIND) service is a distributed database lookup service that allows you to distribute the `hosts` database networkwide. A network running BIND does not have to be connected to the Internet; if it is, however, BIND allows systems on your network to resolve the names and addresses of hosts on the Internet.

BIND is based on a client/server model. Databases are maintained on the primary server, and updated information is distributed to secondary and slave servers. Caching servers have access to the Internet, but do not maintain databases. Instead, they service queries by asking other servers for the information, and then storing the answers they receive. Clients query a server for information. For more information about BIND, and client/server interactions, see the *Network and Communications Overview.*

During the BIND setup and configuration process some or all of the following files are created or modified:

- `/etc/hosts`
- `/etc/named/hosts.db`
- `/etc/named/hosts.rev`
- `/etc/named/named.ca`
- `/etc/named/named.local`
- `/etc/named/named.boot`
- `/etc/rc.config`
- `/etc/resolv.conf`
- `/etc/svc.conf`
- `/var/adm/sendmail/sendmail.cf`

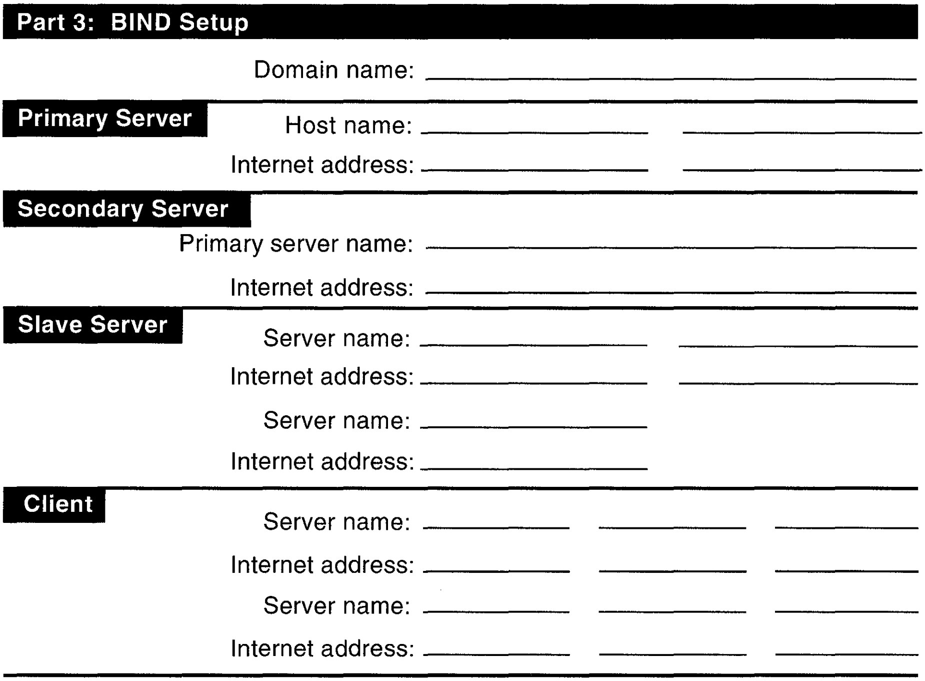
4.1 Gathering Information

Appendix A contains a worksheet that you can use to record the information that you need to complete the tasks in this book. Use Part 3 of the worksheet to record the information you gather as you work your way through this section. To obtain a copy of this worksheet, print the following PostScript file:

`/usr/examples/network_configuration/worksheet.ps`

Figure 4-1 shows Part 3 of the Configuration Worksheet.

Figure 4-1: Configuration Worksheet, Part 3



Part 3: BIND Setup

Domain name: ____________________

Primary Server

Host name: __________ __________

Internet address: __________ __________

Secondary Server

Primary server name: ____________________

Internet address: ____________________

Slave Server

Server name: __________ __________

Internet address: __________ __________

Server name: __________

Internet address: __________

Client

Server name: ________ ________ ________

Internet address: ________ ________ ________

Server name: ________ ________ ________

Internet address: ________ ________ ________

ZK-0762U-R

Gather the following information before setting up a BIND domain:

- The domain name

  Your domain name is assigned by the Network Information Center (NIC) when it assigns your network a number. If your network does not have a number assigned by the NIC, you can create a domain name.

  For information about contacting the NIC, see Section 2.1.

- The role each host will play in your environment

  BIND runs on each system in your network. You must decide what role each system will play within the BIND domain that you are creating. Select one host to be the primary server; there can be only one primary server for each domain. Select one or more hosts to be secondary, slave, and caching servers. The rest of the hosts should run as BIND clients.

- For the primary server, a list of the names and IP addresses of all hosts in the domain
- For secondary servers, the name and IP address of the primary server
- For clients and slave servers, the names and IP addresses of three servers

4.2 Running bindsetup

The following sections describe how to set up a BIND domain by using the `bindsetup` script. With the `bindsetup` script, you can configure clients and servers.

Note that you must set up the primary server first; then, you can configure the other systems in any order.

4.2.1 Setting Up the Primary Server

The primary server runs the `named` daemon and contains the master copy of the `hosts` database. Use the following procedure to set up the primary server:

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

   To create the source file from which the `hosts` database will be created, update the primary server's local `/etc/hosts` file and then copy it into the `/etc/namedb/src` directory. Note that if a system, `host1` for example, is in your BIND domain and is running BIND but is not included in the primary server's `hosts` database, other systems in the domain cannot obtain `host1`'s IP address.

   The format of an entry in the `/etc/hosts` file is as follows:

   IP_address host1 [*alias_1 alias_2 alias_n*]

   Example 4-1 is a sample `/etc/hosts` file.

Example 4-1: Sample /etc/hosts File

```
# @(#)hosts    1.0     (DEC OSF/1)
#
# Description:  The hosts file associates host names with
#               IP addresses.
#
# Syntax:
#    nnn.nnn.nnn.nnn hostname.domain.name [alias_1,...,alias_n] \
#    [#comments]
#
# nnn.nnn.nnn.nnn       The IP address of the host.
# hostname.domain.name  The fully qualified host name, including
#                       the domain name.
```

Example 4-1: (continued)

```
# alias_n                Other names or abbreviations for this
#                        host.
# #comments              Text following the comment character (#)
#                        is ignored.
#
127.0.0.1 localhost
120.105.5.1 host1.cities.dec.com h1
120.105.5.2 host2.cities.dec.com h2
120.105.5.3 host3.cities.dec.com h3     #BIND server
120.105.5.4 host4.cities.dec.com h4     #BIND server
120.105.5.5 host5.cities.dec.com h5
```

Note

> Note that the file that you copy into the /etc/namedb/src directory must be named hosts.

2. Invoke the bindsetup script.

   You can invoke the script either by choosing the Berkeley Internet Name Domain Service (BIND) option from the Setup Menu or by entering the following command:

   ```
   # /usr/sbin/bindsetup
   ```

   An explanation of bindsetup is displayed on your screen.

3. Press Return and choose the a option from the Action Menu.
4. Enter c to continue after the script tells you that you must know your default domain name or exit, and then enter your domain name:

   ```
   Enter the default BIND domain name []: cities.dec.com
   ```

5. Choose the p option from the Configuration Menu, and answer yes when bindsetup asks if you want to convert the source files in /etc/namedb/src to the appropriate BIND format.

   The bindsetup script indicates which system files it is updating, sets the host name to the fully qualified BIND host name, and restarts the Simple Mail Transfer Protocol (SMTP) Mail Service (sendmail).

6. Indicate whether you want bindsetup to start the named daemon.

   If you answer yes, bindsetup starts the daemon.

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

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

7. Indicate the order in which to resolve host name queries.

   This step enables you to choose the order in which to resolve host name queries, as follows:

   - Check the local /etc/hosts database before querying BIND (choose option 1). This is the recommended order.
   - Query BIND first (choose option 2).
   - Run the svcsetup script to customize service order selection (choose option 3).

   If you choose option 3, the bindsetup script invokes the svcsetup script, which allows you to modify the database services selection file (the svc.conf file). See Section 4.2.6 for information on modifying the svc.conf file. Appendix B provides information on editing the svc.conf file with svcsetup or manually.

4.2.2 Setting Up a Secondary Server

Secondary servers run the named daemon and provide backup for the primary server. Secondary servers load their database files from the primary server and periodically poll the primary server to ensure that their databases are up to date. Use the following procedure to set up a secondary server:

1. Invoke the bindsetup script.

   You can invoke the script either by choosing the Berkeley Internet Name Domain Service (BIND) option from the Setup Menu or by entering the following command:

   ```
   # /usr/sbin/bindsetup
   ```

   An explanation of bindsetup is displayed on your screen.

2. Press Return and select the a option from the Action Menu.

3. Enter c to continue after the script tells you that you must know your default domain name or exit, and then enter your domain name:

   ```
   Enter the default BIND domain name []: cities.dec.com
   ```

4. Choose the s option from the Configuration Menu, and enter c after bindsetup explains that you must know the name and IP address of the BIND primary server for your domain.

5. Enter the host name and IP address of the primary server for your domain:

   ```
   Enter the host name of the BIND primary server in
   the "cities.dec.com" domain: host1
   Enter the Internet address for host1.cities.dec.com []:
   120.105.1.26
   ```

   If you enter the fully qualified host name, you must include a trailing dot (.). For example, if the fully qualified host name is `cxcxcx.abc.xyz.com`, you would enter it as follows:

   ```
   cxcxcx.abc.xyz.com.
   ```

   The `bindsetup` script indicates which system files it is updating, sets the host name to the fully qualified BIND host name, and restarts the SMTP Mail Service (`sendmail`).

6. Indicate whether you want `bindsetup` to start the `named` daemon.

   If you answer yes, `bindsetup` starts the daemon.

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

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

7. Indicate the order in which to resolve host name queries.

   This step enables you to choose the order in which to resolve host name queries, as follows:

   - Check the local `/etc/hosts` database before querying BIND (choose option 1). This is the recommended order.
   - Query BIND first (choose option 2).
   - Run the `svcsetup` script to customize service order selection (choose option 3).

   If you choose option 3, the `bindsetup` script invokes the `svcsetup` script, which allows you to modify the database services selection file (the `svc.conf` file). See Section 4.2.6 for information on modifying the `svc.conf` file. Appendix B provides information on editing the `svc.conf` file with `svcsetup` or manually.

4.2.3 Setting Up a Caching Server

Caching servers run the `named` daemon and service queries by asking other servers for the information. They store the information they receive until the data expires. Use the following procedure to set up a caching server:

1. Invoke the `bindsetup` script.

   You can invoke the script either by choosing the Berkeley Internet Name Domain Service (BIND) option from the Setup Menu or by entering the following command:

   ```
   # /usr/sbin/bindsetup
   ```

   An explanation of `bindsetup` is displayed on your screen.

2. Press Return and choose the `a` option from the Action Menu.

3. Enter `c` after the script tells you that you must know your default domain name or exit, and then enter your domain name:

   ```
   Enter the default BIND domain name []: cities.dec.com
   ```

4. Choose the `a` option from the Configuration Menu.

   The `bindsetup` script indicates which system files it is updating, sets the host name to the fully qualified BIND host name, and restarts the SMTP Mail Service (`sendmail`).

5. Indicate whether you want `bindsetup` to start the `named` daemon.

   If you answer yes, the `bindsetup` script starts the daemon.

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

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

6. Indicate the order in which to resolve host name queries.

   This step enables you to choose the order in which to resolve host name queries, as follows:

   - Check the local `/etc/hosts` database before querying BIND (choose option 1). This is the recommended order.
   - Query BIND first (choose option 2).
   - Run the `svcsetup` script to customize service order selection (choose option 3).

   If you choose option 3, the `bindsetup` script invokes the `svcsetup` script, which allows you to modify the database services selection file (the `svc.conf` file). See Section 4.2.6 for information on modifying the `svc.conf` file. Appendix B provides information on editing the `svc.conf` file with `svcsetup` or manually.

4.2.4 Setting Up a Slave Server

Slave servers run the named daemon and forward queries to the list of forwarders specified in their boot file. Caching servers forward queries until the list is exhausted or the query is satisfied. Slave servers store the information they receive until the data expires. Use the following procedure to set up a slave server:

1. Invoke the `bindsetup` script.

   You can invoke the script either by choosing the Berkeley Internet Name Domain Service (BIND) option from the Setup Menu or by entering the following command:

   ```
   # /usr/sbin/bindsetup
   ```

   An explanation of `bindsetup` is displayed on your screen.

2. Press Return and choose the `a` option from the Action Menu.

3. Enter `c` after the script tells you that you must know your default domain name or exit, and then enter your domain name:

   ```
   Enter the default BIND domain name []: cities.dec.com
   ```

4. Choose the `1` option from the Configuration Menu, and enter `c` after the script explains that you must know the names and IP addresses of the specified BIND servers for your domain.

5. Enter three host names and IP addresses of BIND servers for your domain. If the host names and IP addresses are not listed in the `/etc/hosts` file, the `bindsetup` script gives you the option of adding them:

   ```
   Enter the host name of the BIND server in the
         "cities.dec.com" domain: host1

   Enter the Internet address for host1.cities.dec.com
        [120.105.1.26] Return

   Enter the host name of the BIND server in the
        "cities.dec.com" domain: host2

   Enter the Internet address for host2.cities.dec.com []:
         120.105.1.27

   Would you like to add host2 to the /etc/hosts
         file (y/n) [n] ? y
   Enter the host name of the BIND server in the
        "cities.dec.com" domain: host3

   Enter the Internet address for host3.cities.dec.com []:
         120.105.1.28

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

   If you enter the fully qualified host name, you must include a trailing dot (.). For example, if the fully qualified host name is cxcxcx.abc.xyz.com, you would enter it as follows:

```
cxcxcx.abc.xyz.com.
```

6. Indicate that you are finished entering BIND servers.

   The bindsetup script indicates which system files it is updating, sets the host name to the fully qualified BIND host name, and restarts the SMTP Mail Service (sendmail).

7. Indicate whether you want bindsetup to start the named daemon.

   If you answer yes, bindsetup starts the daemon.

   If you answer no, you must start the daemon manually after bindsetup exits and returns you to the system prompt (#):

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

8. Indicate the order in which to resolve host name queries.

   This step enables you to choose the order in which to resolve host name queries, as follows:

   - Check the local /etc/hosts database before querying BIND (choose option 1). This is the recommended order.
   - Query BIND first (choose option 2).
   - Run the svcsetup script to customize service order selection (choose option 3).

   If you choose option 3, the bindsetup script invokes the svcsetup script, which allows you to modify the database services selection file (the svc.conf file). See Section 4.2.6 for information on modifying the svc.conf file. Appendix B provides information on editing the svc.conf file with svcsetup or manually.

4.2.5 Setting Up a Client

BIND clients query servers for host name and address information. They do not run the named daemon. Use the following procedure to set up a client system:

1. Invoke the bindsetup script.

   You can invoke the script either by choosing the Berkeley Internet Name Domain Service (BIND) option from the Setup Menu or by entering the following command:

```
# /usr/sbin/bindsetup
```

An explanation of `bindsetup` is displayed on your screen.

2. Press Return and choose the `a` option from the Action Menu.
3. Enter `c` after the script tells you that you must know your default domain name or exit, and then enter your domain name:

```
Enter the default BIND domain name []: cities.dec.com
```

4. Choose the `c` option from the Configuration Menu, and enter `c` after the script explains that there must be at least one BIND primary or secondary server configured for your domain and that you must know the names and IP addresses of the specified BIND servers for your domain.
5. Enter three host names and IP addresses of BIND servers for your domain. The addresses are placed in the `/etc/resolv.conf` file, where the resolver uses them to determine the IP addresses of name servers it should query. If the host names and IP addresses are not listed in the `/etc/hosts` file, the `bindsetup` script gives you the option of adding them:

```
Enter the host name of the BIND server in the
      "cities.dec.com" domain: host1

Enter the Internet address for host1.cities.dec.com []:
      120.105.1.26

Would you like to add host1 to the /etc/hosts
      file (y/n) [n] ? y

Enter the host name of the BIND server in the
     "cities.dec.com" domain: host2

Enter the Internet address for host2.cities.dec.com
     [120.105.1.27]: Return

Enter the host name of the BIND server in the
      "cities.dec.com" domain: host3

Enter the Internet address for host3.cities.dec.com []:
      120.105.1.28

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

If you enter the fully qualified host name, you must include a trailing dot (.). For example, if the fully qualified host name is `cxcxcx.abc.xyz.com`, you would enter it as follows:

```
cxcxcx.abc.xyz.com.
```

6. Indicate that you are finished entering BIND servers.

The `bindsetup` script indicates which system files it is updating, sets the host name to the fully qualified BIND host name, and restarts the SMTP Mail Service (`sendmail`).

7. Indicate the order in which to resolve host name queries.

   This step enables you to choose the order in which to resolve host name queries, as follows:

   - Check the local `/etc/hosts` database before querying BIND (choose option 1). This is the recommended order.
   - Query BIND first (choose option 2).
   - Run the `svcsetup` script to customize service order selection (choose option 3).

   If you choose option 3, the `bindsetup` script invokes the `svcsetup` script, which allows you to modify the database services selection file (the `svc.conf` file). See Section 4.2.6 for information on modifying the `svc.conf` file. Appendix B provides information on editing the `svc.conf` file with `svcsetup` or manually.

4.2.6 Modifying the svc.conf File with svcsetup

While running the `bindsetup` script, you are given the option of editing the `/etc/svc.conf` file with the `svcsetup` script. If you choose this option, the `bindsetup` script invokes the `svcsetup` script. Use the following procedure to edit the `/etc/svc.conf` file:

1. Press Return following the informational messages to continue.
2. Press Return to choose the `m` option from the Configuration Menu.
3. Choose 2 from the Change Menu.

   The number 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 system will be searched first for the requested information. If the information is not found locally, then BIND servers, NIS servers, or both, are queried, depending on which options you choose.

   Note

   > Digital recommends that `local` be the first service that your system queries for all databases, regardless of what services you are running.

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

The svcsetup script indicates that it is updating the /etc/svc.conf file. Both svcsetup and bindsetup indicate that they have completed and you are returned to the system prompt (#).

Setting Up the Network Information Service 5

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.

NIS is based on a client/server model. Database files, or maps, are located in the `/var/yp/`*`domainname`* directory, and are stored and maintained on a master server. Changes to the database files are propagated at regular intervals to the slave servers. Clients do not store databases locally; they query servers for information. For more information about NIS and client/server interactions, see the *Network and Communications Overview.*

By default, NIS distributes the `aliases`, `group`, `hosts`, `mail.aliases`, `netgroup`, `networks`, `passwd`, `protocols`, `rpc`, and `services` databases. (The `mail.aliases` and `netgroup` database are created exclusively for NIS.) You can also create and distribute site-specific customized databases, such as NFS `automount` maps. For information on creating `automount` maps for distribution by NIS, see Appendix D. For information on creating and distributing other site-specific NIS maps, see the *Network Administration and Problem Solving* manual.

During the NIS setup and configuration process some or all of the following files are created or modified:

- Database maps (for master server only)
- Domain directory (for servers only)
- `/etc/rc.config`
- `/etc/svc.conf`
- `/var/yp/Makefile` (for master server only)
- `/var/spool/cron/crontabs/root` (for slave servers only)
- `/var/yp/src/mail.aliases` (for master server only)
- `/var/yp/src/netgroup` (for master server only)
- `/etc/passwd`
- `/etc/group`

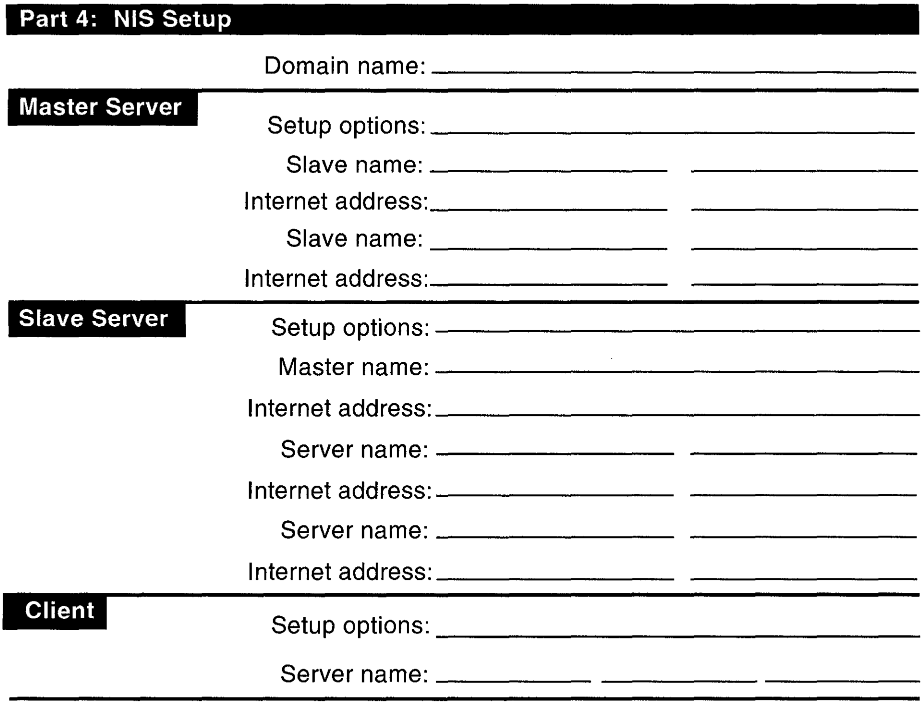
5.1 Gathering Information

Appendix A contains a worksheet that you can use to record the information that you need to complete the tasks in this book. Use Part 4 of the worksheet to record the information you gather as you work your way through this section. To obtain a copy of this worksheet, print the following PostScript file:

```
/usr/examples/network_configuration/worksheet.ps
```

Figure 5-1 shows Part 4 of the Configuration Worksheet.

Figure 5-1: Configuration Worksheet, Part 4



Part 4: NIS Setup

Domain name: ____________________

Master Server

Setup options: ____________________

Slave name: __________ __________

Internet address: __________ __________

Slave name: __________ __________

Internet address: __________ __________

Slave Server

Setup options: ____________________

Master name: ____________________

Internet address: ____________________

Server name: __________ __________

Internet address: __________ __________

Server name: __________ __________

Internet address: __________ __________

Client

Setup options: ____________________

Server name: ______ ______ ______

ZK-0763U-R

Gather the following information before setting up an NIS domain:

- The domain name

  An NIS domain is an administrative entity that is organized into a master server, one or more slave servers, and numerous clients. All systems in a domain share the same set of NIS database files. The domain name that you choose can be any string of 31 or fewer alphanumeric characters. All systems in the domain must declare the same domain name.

- The role each host will play in your distributed environment

  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 should run as NIS clients.

  Note

  > The master server and all slave servers are also considered to be NIS clients.

- For the master server, a complete list of the names and IP addresses of slave servers in the domain
- For the master server, whether to run the `yppasswdd` daemon

  The `yppasswdd` daemon runs on the master server and allows the master copy of the password file to be updated remotely using the `yppasswd` command. Digital recommends that you run the `yppasswdd` daemon.

- For slave servers, the host name and IP address of the master server
- For all systems, whether you want to lock the `ypbind` daemon to a particular domain name and server list

  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. Digital recommends that you run NIS with the `–S` option configured.

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

- For all systems, whether you want to run NIS with the `–ypset` option, the `–ypsetme` option, or with both options set

  The `–ypset` option allows a user running 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. Digital recommends that you run NIS with neither the `–ypset` nor the `–ypsetme` options.

- Whether clients in the domain will use the `automount` program

  The `automount` program is an alternative to mounting remote file systems, allowing users to mount remote file systems on an as-needed basis. When NIS is used to distribute `automount` maps, creating and administering the maps for the NIS domain is the responsibility of the administrator of the NIS master server. For information on creating and administering automount maps, see Appendix D. For information on administering automount maps, see Section 6.3.1.

Whether you use the automount program depends on your site's networking environment.

5.2 Running nissetup

The following sections describe how to set up NIS, using the nissetup script. With the nissetup script, you can configure servers and clients.

Note that you must set up the master server first. After the master server is set up, you can configure the other systems in any order.

5.2.1 Setting Up the Master Server

The master copies of the databases being served by NIS reside on the master server. Note that you must perform steps 1 through 5 of the following procedure before running the nissetup script.

To set up the master server, log in as superuser and perform the following steps:

1. Copy into the /var/yp/src directory the local /etc files that you intend to make into NIS maps for distribution and make sure that all of the information in them is up to date. These files usually include the following:

   - aliases
   - group
   - hosts
   - networks
   - passwd
   - protocols
   - rpc
   - services

   If you do not want to distribute one of these default maps, do not copy the local /etc file for it into the /var/yp/src directory. If a file is absent from the /var/yp/src directory while it is building the default NIS maps, the nissetup command 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. Create the /var/yp/src/mail.aliases file.

   The mail.aliases file defines networkwide mail aliases. Creating this file is optional. However, if you want to define and distribute mail aliases on your network, you must create it. If you choose not to create a mail.aliases file, while it is building the NIS maps, the nissetup command issues an informational message that it could not find the mail.aliases file.

   For information on defining mail aliases, see the aliases(4) reference page.

3. Create the /var/yp/src/netgroup file.

   The netgroup file defines networkwide groups and is used for permission checking when doing remote mounts, remote logins, and remote shells. Creating this file is optional. However, if you want to define and distribute netgroup information on your network, you must create the file. If you choose not to create a netgroup file, while it is building the NIS maps, the nissetup command issues an informational message that it could not find the netgroup file.

   For information on defining network groups, 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 automount maps, you must remove the comment sign (#) from the beginning of each of the following lines. These lines were added to the Makefile for the automount daemon.

```
   •
   •
   •
#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/#.*$$// $(DIR)/auto.home | \
#              $(MAKEDBM) - $(YPDBDIR)/$(DOM)/auto.home; \
#              $(TOUCH) $(YPDBDIR)/$(DOM)/auto.home.time; \
#              $(ECHO) "updated auto.home"; \
#              if [ ! $(NOPUSH) ]; then \
#                      $(YPPUSH) auto.home; \
#                      $(ECHO) "pushed auto.home"; \
#              else \
#                      : ; \
#              fi \
#       else \
#               $(ECHO) "couldn't find $(DIR)/auto.home"; \
#       fi
#
#$(YPDBDIR)/$(DOM)/auto.master.time: $(DIR)/auto.master
#        -@if [ -f $(DIR)/auto.master ]; then \
#             $(SED) -e "/^#/d" -e s/#.*$$// $(DIR)/auto.master | \
#             $(MAKEDBM) - $(YPDBDIR)/$(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 an entry for them to the `Makefile`. See the *Network Administration and Problem Solving* manual for information on adding entries for site-specific NIS maps, other than the `/etc/auto.master` and `/etc/auto.home` `automount` maps, to the `/var/yp/Makefile` file.

5. Copy the `automount` maps, or any other site-specific maps, to the `/var/yp/src` directory.

   For information on creating `automount` maps, see Appendix D. For information on creating other site-specific maps, see the *Network Administration and Problem Solving* manual.

6. Invoke the `nissetup` script.

   You can invoke `nissetup` either by choosing the Network Information Service (NIS) option from the Setup Menu or by entering the following

command:

```
# /usr/sbin/nissetup
```

A message is displayed reminding 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.

7. An explanation of `nissetup` is displayed on your screen. 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.

8. Enter and confirm your system's NIS domain name.

9. Choose option 1 to indicate that you are configuring the master server:

```
Will host1 be a

              1. MASTER server,
              2. SLAVE server, or
              3. CLIENT ?

1 2 or 3 [3] ? 1
```

10. Following the `nissetup` script's explanation that there can be only one master server configured for each NIS domain, enter `c` and indicate whether you want to run the `yppasswdd` daemon.

    Digital recommends that you run the `yppasswdd` daemon on the master NIS server.

11. Enter the names of hosts that will be configured as slave servers for this domain.

    If you enter the name of a host 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 and gives you the option to redo it to correct errors or to 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.
```

12. Indicate whether you want to use the –s security option.

    If you choose to run the –s option, you must enter the names of up to four NIS servers.

    The nissetup script automatically places the host name of the server you are configuring first. Press Return when you are done entering server names.

    ```
    Server 1 name: host1
       (An NIS server must specify itself FIRST)
    Server 2 name: host2
    Server 3 name: host3
    Server 4 name: Return
    ```

    Digital recommends that you use the –s option.

13. Indicate whether you want to allow ypset requests on your system.

    Digital recommends that you disallow all ypset requests. Press Return to accept the default, and confirm your choice.

14. Indicate whether you want your system to use all of the NIS databases served by the master server.

    Digital recommends that you use all of the NIS databases.

    If you choose to use all of the NIS databases (by either entering y or accepting the default), 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 18.

    If you choose not to use all of the NIS databases (by entering n), continue with the next step.

15. Indicate whether you want to add a plus sign followed by a colon (+:) to the end of the local /etc/passwd and /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 now 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 all that is required for your system to search NIS.

16. Indicate whether you want to use NIS to obtain information for all of the default databases (other than the /etc/passwd and /etc/group which were dealt with in step 15).

    If you answer yes, nissetup edits the svc.conf file to inclue the string yp for each database. The nissetup script then skips the next question and continues at step 18.

    If you answer no, nissetup continues with the next question.

17. Indicate whether you want the nissetup script to invoke the svcsetup script. (Note, if you answered yes to step 16, skip this step.)

    If you answer yes, nissetup invokes the svcsetup script, which allows you to modify the database services selection file (the svc.conf file). See Section 5.2.4 for information on modifying the svc.conf file.

    If you answer no, nissetup continues with the next question. Note that you must edit the svc.conf file if you want your system to use NIS to obtain database information other than passwd and group information. See Appendix B for information on editing the svc.conf file with svcsetup or manually.

18. Indicate whether 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
    ```

5.2.2 Setting Up a Slave Server

Slave servers obtain copies of their domain's NIS maps from the master server. Their maps are updated periodically over the network. If the master server goes down, the flow of database information throughout the domain is sustained by the slave servers. Use the following procedure to set up a slave server:

1. Invoke the `nissetup` script.

   You can invoke `nissetup` either by choosing the Network Information Service (NIS) option from the Setup Menu or by entering the following command:

   ```
   # /usr/sbin/nissetup
   ```

   A message is displayed reminding 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.

2. An explanation of `nissetup` is displayed on your screen. 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.

3. Enter and confirm your system's NIS domain name.

4. Choose option 2 to indicate that you are configuring a slave server:

   ```
   Will host2 be a

                  1. MASTER server,
                  2. SLAVE server, or
                  3. CLIENT ?
   1 2 or 3 [3] ? 2
   ```

5. 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.

6. Enter the name of the master server for your domain.

7. Indicate whether you want to use the `–S` security option.

   If you choose to run the `–S` option, you must enter the names of up to four NIS servers.

   The `nissetup` script automatically places the host name of the server you are configuring first. Press Return when you are finished entering server names.

   ```
   Server 1 name: host2
      (An NIS server must specify itself FIRST)
   Server 2 name: host1
   Server 3 name: host3
   Server 4 name: Return
   ```

   Digital recommends that you use the `–S` option.

   If you enter the name of a host that is not listed in the slave server's `/etc/hosts` file, the `nissetup` script prompts you for its IP address.

When you are done entering the list of servers, enter c to continue configuring NIS on your system.

8. Indicate whether you want to allow ypset requests on your system.

   Digital recommends that you disallow all ypset requests. Press Return to accept the default, and confirm your choice.

9. Indicate whether you want your system to use all of the NIS databases served by the master server.

   Digital recommends that you use all of the NIS databases.

   If you choose to use all of the NIS databases (by either entering y or accepting the default), 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 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 (by entering n), continue with the next step.

10. Indicate whether you want to add +: to the end of the local /etc/passwd and /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 now 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 all that is required for your system to search NIS.

11. Indicate whether you want to use NIS to obtain information for all of the default databases.

    If you answer yes, nissetup edits the svc.conf file to include the string yp for each database. The nissetup script then skips the next question and continues at step 13.

12. Indicate whether you want the nissetup script to invoke the svcsetup script. (Note, if you answered yes to step 11, skip this step.)

    If you answer yes, nissetup invokes the svcsetup script, which allows you to modify the database services selection file (the svc.conf file). See Section 5.2.4 for information on modifying the svc.conf file.

If you answer no, nissetup continues with the next question. Note that you must edit the svc.conf file if you want your system to use NIS to obtain database information other than passwd and group information. See Appendix B for information on editing the svc.conf file with svcsetup or manually.

13. Indicate whether 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
    ```

5.2.3 Setting Up a Client

NIS clients query servers for database information. They do not maintain copies of the NIS maps for their domain. Use the following procedure to set up a client:

1. Invoke the nissetup script.

   You can invoke nissetup either by choosing the Network Information Service (NIS) option from the Setup Menu or by entering the following command:

   ```
   # /usr/sbin/nissetup
   ```

   A message is displayed reminding 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.

2. An explanation of nissetup is displayed on your screen. 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.

3. Enter and confirm your system's NIS domain name.

4. Press Return to accept the default that you are configuring a client:

   ```
   Will host5 be a

                 1. MASTER server,
                 2. SLAVE server, or
                 3. CLIENT ?
   1 2 or 3 [3] ? Return
   ```

5. Enter c to continue following the nissetup script's warning that at least one server must be configured for this domain.

6. Indicate whether you want to use the –S security option.

   If you choose to run the –S option, you must enter the names of up to four NIS servers.

   If you enter the name of a host that is not listed in the client's /etc/hosts file, the nissetup script prompts you for its IP address. After you have completed entering the list of servers, enter c to continue configuring NIS on your system.

7. Indicate whether you want to allow ypset requests on your system.

   Digital recommends that you disallow all ypset requests. Press Return to accept the default, and confirm your choice.

8. Indicate whether you want your system to use all of the NIS databases served by the master server.

   Digital recommends that you use all of the NIS databases.

   If you choose to use all of the NIS databases (by either entering y or accepting the default), 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 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 12.

   If you choose not to use all of the NIS databases (by entering n), continue with the next step.

9. Indicate whether you want to add +: to the end of the local /etc/passwd and /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 now 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 all that is required for your system to search NIS.

10. Indicate whether you want to use NIS to obtain information for all of the default databases.

    If you answer yes, nissetup edits the svc.conf file to include the string yp for each database. The nissetup script then skips the next question and continues at step 12.

If you answer no, nissetup continues with the next question.

11. Indicate whether you want the nissetup script to invoke the svcsetup script. (Note, if you answered yes to step 10, skip this step.)

    If you answer yes, nissetup invokes the svcsetup script, which allows you to modify the database services selection file (the svc.conf file). See Section 5.2.4 for information on modifying the svc.conf file.

    If you answer no, nissetup continues with the next question. Note that you must edit the svc.conf file if you want your system to use NIS to distribute database information other than passwd and group information. See Appendix B for information on editing the svc.conf file with svcsetup or manually.

12. Indicate whether 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
    ```

5.2.4 Modifying the svc.conf File with svcsetup

If you choose not to use NIS for all of the default databases, the nissetup script provides the option of editing 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 running on your system.

   The default choice (2) indicates that the local /etc files will be searched first for the requested information. If the information is not found locally, then an NIS server will be 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 BIND, choose either option 5 or 6 for the `hosts` database. Note that options 3, 4, 5, and 6 are valid for the `hosts` database only.

Setting Up the Network File System 6

The Network File System (NFS) is a facility for sharing files in a heterogeneous environment. It is based on the client/server model where an NFS server is a system that exports file systems, and an NFS client is a system that imports file systems. A client can mount file systems by using either the `/etc/fstab` file or the `automount` daemon. Both setup methods are explained in this chapter.

Your system can be set up as an NFS server, an NFS client, or both. For more information about NFS and about client/server interactions, see the *Network and Communications Overview.*

During the NFS setup and configuration process the following files are created or modified:

- `automount` maps (if you are using the `automount` daemon)
- `/etc/exports`
- `/etc/fstab` (if you are using `/etc/fstab`)
- `/etc/rc.config`

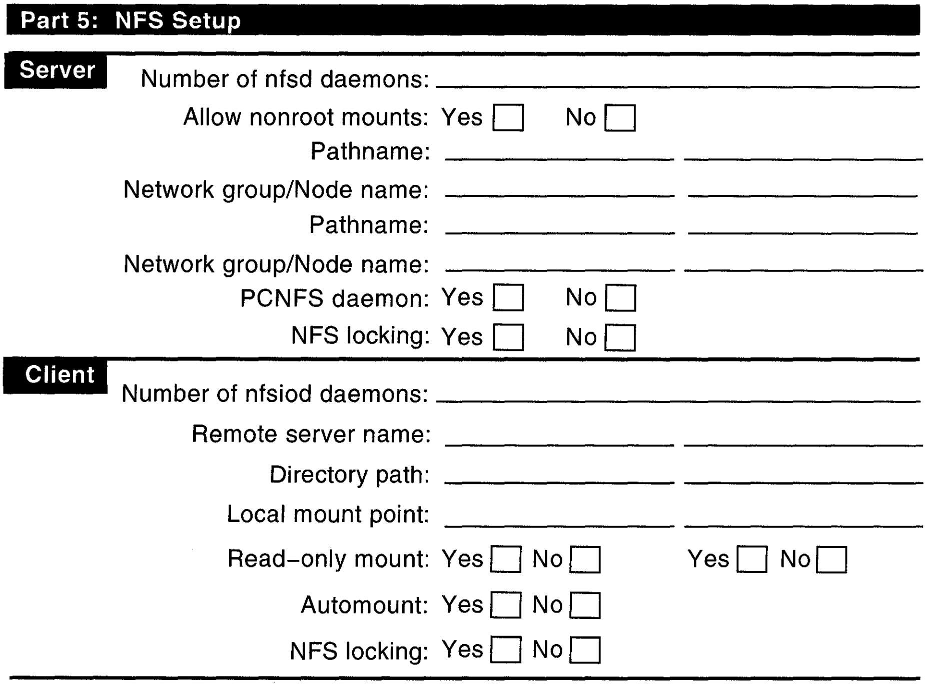
6.1 Gathering Information

Appendix A contains a worksheet that you can use to record the information that you need to complete the tasks in this book. Use Part 5 of the worksheet to record the information you gather as you work your way through this section. To obtain a copy of the worksheet, print the following PostScript file:

```
/usr/examples/network_configuration/worksheet.ps
```

Figure 6-1 shows Part 5 of the Configuration Worksheet.

Figure 6-1: Configuration Worksheet, Part 5



Part 5: NFS Setup

Server

Number of nfsd daemons: ____________________

Allow nonroot mounts: Yes □ No □

Pathname: __________ __________

Network group/Node name: __________ __________

Pathname: __________ __________

Network group/Node name: __________ __________

PCNFS daemon: Yes □ No □

NFS locking: Yes □ No □

Client

Number of nfsiod daemons: ____________________

Remote server name: __________ __________

Directory path: __________ __________

Local mount point: __________ __________

Read-only mount: Yes □ No □ Yes □ No □

Automount: Yes □ No □

NFS locking: Yes □ No □

ZK-0764U-R

Gather the following information and make the following decisions before setting up NFS:

- Whether your system will be an NFS server, an NFS client, or both
- For servers:
  - The number of `nfsd` daemons to run

    The default number of 8 is adequate for an average work load. If you use `nfssetup` to set up NFS on your system, you can configure from 0 to 128 `nfsd` daemons. You can start additional `nfsd` daemons from the command line. See the `nfsd`(8) reference page for information on starting `nfsd` daemons from the command line.
  - The pathnames of the file systems or directories that you intend to export
  - The permissions that you want 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. For more information on assigning permissions to exported file systems or directories and on specifically mapping the `root` UID for clients, see the `exports`(4) reference page.

  - The network groups or individual host names to which you will export these file systems or directories

    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. For information on defining network groups, see the the `netgroup`(4) reference page.

  - Whether to allow nonroot mounts

    If you allow nonroot mounts (by setting the `NONROOTMOUNTS` parameter to 1), 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.

- For clients:

  - The number of block I/O (`nfsiod`) daemons to run

    The default number of 7 is recommended for optimum load generation on DEC OSF/1 servers. If you use `nfssetup` to set up NFS on your system, you can configure from 0 to 20 `nfsiod` daemons. You can start `nfsiod` daemons from the command line. See the `nfsiod`(8) reference page for information on starting `nfsiod` daemons from the command line.

  - The remote host names (servers) from which you are importing file systems or directories

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

  - The local mount points where you want the imported file systems or directories to reside

  - The permissions for the imported file systems or directories

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.

- For clients using the `automount` daemon, determine if the network is running the Network Information Service (NIS)

  You can set up `automount` maps on the local system, but if the network is running NIS, the `automount` maps are better administered and served from the master NIS server. The format of the maps is the same whether they are local or served by the NIS master server. For information on creating `automount` maps, see Appendix D.

6.2 Running nfssetup

The following steps describe how to set up NFS by using the `nfssetup` script. With the `nfssetup` script, you can configure both NFS servers and clients.

Note

> If your network is running NIS or Berkeley Internet Name Domain (BIND) 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 BIND database.
>
> Similarly, if your network is running NIS or BIND 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.

1. Invoke the `nfssetup` script either by choosing the Network File System (NFS) option from the Setup Menu or by entering the following

command:

```
# /usr/sbin/nfssetup
```

The script prompts you for information about your system.

2. Indicate whether you want to enable NFS locking.

   If you enable locking, the NFS lock manager (rpc.lockd) and the status monitor (rpc.statd) are run. Running these deamons allows users to use fcntl(2) and lockf(3) to lock file regions on NFS files (in addition to local files). Not running the daemons means that users can only use advisory locking primitives on local files. By default, the script runs the daemons.

3. Indicate whether your system will export directories.

4. If you answered yes in step 3, nfssetup asks you whether your system will allow nonroot mounts.

5. If you answered yes in step 3, nfssetup prompts you for the number of nfsd daemons to run.

6. Indicate the number of block I/O (nfsiod) daemons to run.

7. Indicate whether you want to run the PC-NFS (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. Also, you must export the /usr/spool/pcnfs directory to the PC client to enable the client to utilize network printing. For information on exporting directories, see the *Network Administration and Problem Solving* manual.

8. Indicate whether you want to run the automount daemon.

   If you answer yes, go to the next step.

   If you answer no, go to step 10.

   For more information, see Section 6.3 and Appendix D.

9. Specify the argument list to pass to automount(8).

   Note that you can later change the automount(8) argument list by using a rcmgr command to set the AUTOMOUNT_ARGS variable.

   For more information, see automount(8) and rcmgr(8).

10. If you choose to export directories, nfssetup prompts you for the full pathname of the directory to be exported and the names of the hosts or network groups allowed to import the directory. If you do not specify individual hosts or network groups, all hosts on the network can import the file system. Press Return to indicate that you are finished entering

information:

```
Enter the directory pathname: /usr/var/tmp
     Netgroup/Machine name: host1
     Netgroup/Machine name: host2
     Netgroup/Machine name: Return

Enter the directory pathname: Return
Directory export list complete...
```

11. If your system is importing directories, enter the host name of the system from which you are importing the directory, its full pathname, the local mount point, and whether it is a read-only mount. If the local mount point does not exist, `nfssetup` creates it.

```
Enter the remote host name: rhost1

   Enter the remote directory pathname: /usr/share/man
   Enter the local mount point: /usr/share/man
   Is this a read-only mount [y] ? Return
   Enter the remote directory pathname: Return
Enter the remote host name: Return
Remote directory mount list complete...
```

Note

If you place NFS mount points to more than one server in a given directory, the `getwd` routine sometimes blocks on an attempt to obtain the pathname of the current working directory.

When computing the pathname string, the `getwd` routine moves up the tree from the current working directory to the `root` and calls the `readdir` routine at each level to obtain a pointer to the next directory level. When `getwd` passes through a mount point, the routine uses the `stat` system call to process all entries in the directory until information for the mount point just traversed is returned. If a directory entry is a mount point to a different server and that server is hard mounted and down, the `stat` system call keeps trying to access the directory until its server is able to respond. As a result, the calling `getwd` routine blocks (waits for return status) until the server is available and can respond to the `stat` call. To avoid this problem with the `getwd` routine, place mount points to different servers in separate directory trees. Some directories (such as `/usr`) in complex production environments might be too large for you to adhere strictly to this recommendation. In such cases, try to minimize the number of mount points to different servers that occur in any given directory.

12. Enter c to confirm the information that you entered, if it is correct. If it is incorrect, enter r and redo it.

    The nfssetup script indicates what system files it is updating.

13. Indicate whether you want to start the NFS daemons immediately.

    If you answer yes, nfssetup starts the daemons. If you answer no, enter the following command to start the daemons manually after nfssetup exits and returns you to the system prompt (#):

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

14. To mount the remote directories listed in your /etc/fstab file without rebooting the system, enter the following command:

    ```
    # mount -a -t nfs
    ```

6.3 Using automount to Set Up Clients

The automount daemon offers an alternative 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 using the automount daemon invokes a command that needs to access a remotely mounted file or directory, the automount daemon mounts that file system or directory and keeps it mounted for as long as the user is using it. When a specified amount of time elapses (the default is 5 minutes) without the file system or directory being accessed, the automount daemon unmounts it.

Use the following procedure to set up a client to use the automount daemon:

1. Create automount maps.

   The automount maps indicate which remote file systems the automount daemon monitors, where they should be mounted, and with what mount options.

   Typically, NIS is used to distribute automount maps. The system administrator on the NIS master server maintains the maps that are distributed by NIS. For information on how the administrator of the NIS master server builds and distributes automount maps, see Section 5.2.1.

   The specifics of writing automount maps are discussed in Appendix D.

2. Start the automount daemon.

   You can start the automount daemon by running the nfssetup script (see Section 6.2). If you start the automount daemon by running the nfssetup script, you do not have to do step 3.

Alternatively, you can add the AUTOMOUNT flag to the /etc/rc.config file to indicate that the system is using the automount daemon.

Edit the /etc/rc.config file by using the /usr/sbin/rcmgr utility, as follows:

```
# /usr/sbin/rcmgr set AUTOMOUNT 1
```

To specify options to the automount command, use the rcmgr utility to set the AUTOMOUNT_ARGS variable in the /etc/rc.config file. For example, you can indicate the location of a local master map as follows:

```
# /usr/sbin/rcmgr set AUTOMOUNT_ARGS "-f /etc/auto.master"
```

3. Start the NFS daemons by entering the following command:

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

If the NFS daemons are already running on your system, you must stop and restart them by enter the following commands:

```
# /sbin/init.d/nfs stop
# /sbin/init.d/nfs start
```

Starting the automount daemon without any options or arguments should be adequate for NIS clients running in an environment where an auto.master and other automount maps are set up on the NIS master. If you run the automount daemon locally, or if you want to further customize your automount daemon setup, you should specify some options to the automount daemon when you start it up. For information about invoking the automount daemon and its options, see automount(8) and Section 6.3.2.

6.3.1 Administering automount Maps

You can customize automount maps to suit your environment and administer them in several ways:

- You can use NIS to create and distribute the automount maps.
- You can administer the automount maps locally.
- You can use a combination of both methods.

6.3.1.1 Using NIS to Administer automount Maps

NIS allows you to create and distribute customized maps and, typically, is used to distribute automount maps. Therefore, if NIS is used on your network to distribute automount maps, your system must be an NIS client. When NIS is used to distribute automount maps, the administrator of the

NIS master server creates and administers the maps for the NIS domain.

If many clients in an environment remotely mount a 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 automount 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 automount map changes, or its server changes, the administrator of automount maps 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 Chapter 5 for information on configuring a master NIS server to serve automount maps.

6.3.1.2 Administering automount Locally

Local automount 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 an automount map.

Administering the automount daemon locally is the same as administering it when NIS distributes the maps, except that you, as administrator of your system, create and manage automount maps.

A local auto.master map serves the same function as one distributed in an NIS domain. If a local auto.master is specified, the automount 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 automount daemon is invoked correctly.

6.3.2 Invoking automount

You can specify instructions for the automount daemon 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 automount daemon reads and carries out its 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 daemon without any options, it looks for a distributed NIS map called auto.master. If it finds one, it checks the master map for information about the location of other maps, their local mount points, and the mount options. If it does not find one, and if no local auto.master is specified, the automount daemon exits.

You can invoke the automount daemon from the command line or from an entry in the /etc/rc.config file in one of the following ways:

- Specify all of the arguments to the automount command on the command line. For example:

  ```
  # automount /net –hosts \
    /home /etc/auto.home -rw,intr \
    /- /etc/auto.direct -ro,intr
  ```

- Include the previous information 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 automount at the superuser prompt (#) produces the same results as the previous command line.

- Include the automount command information in a local auto.master file and use the –f option to instruct the automount daemon to consult the local auto.master file first for instructions. The –f option instructs the automount daemon to consult the local master map first and then the NIS-distributed master map. (The –m option instructs the automount daemon to ignore the NIS-distributed master map completely, if there is one.) For example:

  ```
  # automount –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
```

- 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
```

See `automount`(8) for more information on the `automount` command and its options.

Setting Up the UNIX-to-UNIX Copy Program 7

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.

DEC OSF/1 implements the HoneyDanBer version of UUCP. For general information about UUCP see the *Network and Communications Overview*. For information on how to use UUCP, see the *Command and Shell User's Guide* manual.

While setting up and configuring UUCP, some or all of the following files are created or modified:

- `/etc/inittab`
- `/etc/passwd`
- `/usr/lib/uucp/Devices`
- `/usr/lib/uucp/Systems`
- `/usr/lib/uucp/Dialcodes`
- `/usr/lib/uucp/Permissions`
- `/usr/lib/uucp/Poll`
- `/etc/inetd.conf`

7.1 Required Hardware Configurations

You must have one of the following hardware configurations to operate UUCP:

- A Digital modem with Automatic Calling Unit (ACU), as listed in the Software Product Description (SPD) included in your media kit
- A direct connection with a null modem cable such as a BC03-M
- A connection with a modem link

To connect a Digital modem do the following:

1. Connect the modem to a port on the local system by using a straight-through cable.

2. Connect the modem to the phone line by following the instructions in the user's guide supplied with your modem.
3. Set the modem's communications baud rate; see the switch options in the modem's user's guide.

The `tip` command establishes a full-duplex connection to another system. For it to work, the modems must be set up properly for UUCP.

To install a hardwired direct link, connect a null modem cable from a port on the local system to a port on the remote system.

A successful connection between modems requires that both the local and remote modems be 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 Chapter 17.

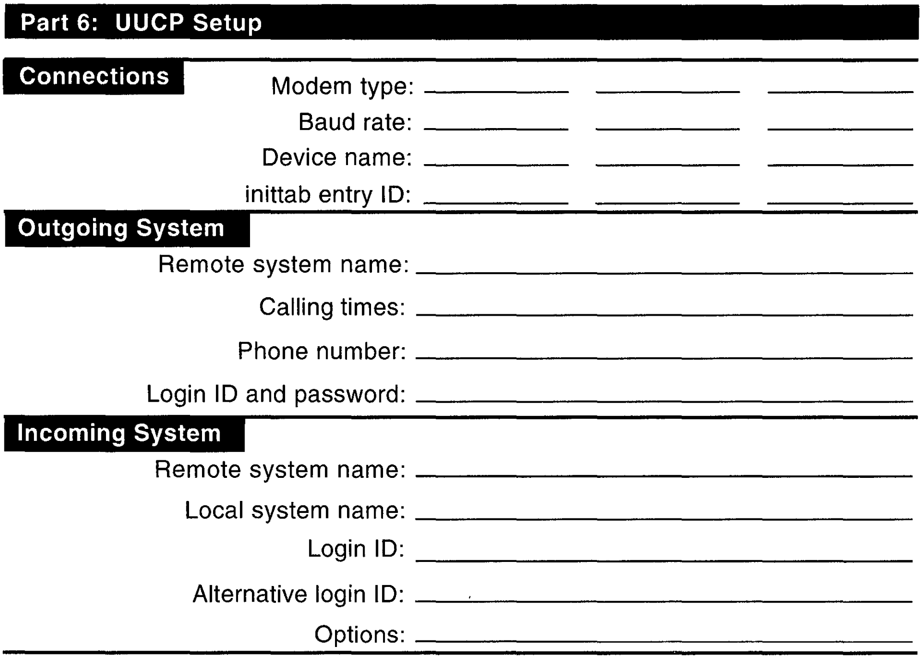
7.2 Gathering Information

Appendix A contains a worksheet that you can use to record the information that you need to complete the tasks in this book. Use Part 6 of the worksheet to record the information you gather as you work your way through this section. To obtain a copy of this worksheet, print the following PostScript file:

```
/usr/examples/network_configuration/worksheet.ps
```

Figure 7-1 shows Part 6 of the Configuration Worksheet.

Figure 7-1: Configuration Worksheet, Part 6



Part 6: UUCP Setup

Connections

Modem type: __________ __________ __________

Baud rate: __________ __________ __________

Device name: __________ __________ __________

inittab entry ID: __________ __________ __________

Outgoing System

Remote system name: ______________________________

Calling times: ______________________________

Phone number: ______________________________

Login ID and password: ______________________________

Incoming System

Remote system name: ______________________________

Local system name: ______________________________

Login ID: ______________________________

Alternative login ID: ______________________________

Options: ______________________________

ZK-0768U-R

Gather the following information before setting up UUCP:

- Be certain that the correct hardware is installed.

  There must be a communications link between two systems before UUCP can work. To use UUCP, the correct hardware must be installed and the UUCP software must be configured to reflect the hardware used. See Section 7.1 for more information about required hardware configurations.

- For configuring communications links you need to know:
  - Modem link
    - Modem type
    - Baud rate
    - Device name
    - ID for `/etc/inittab` entry
  - Direct hardwired link
    - Baud rate

    - Device name
    - ID for /etc/inittab entry
  - For outgoing systems you need to know:
    - System name of each remote system
    - Mode of connection for each remote system
    - Calling times for each remote system
    - Phone number for each remote system
    - Login ID and password for each remote system
  - For incoming systems you need to know:
    - System name of each remote system
    - Local system name
    - Login ID for each remote system
    - Alternative login ID for each remote system (if any)
    - Options for each remote system

You must supply the administrator of each remote system that will call your system with the following information:

- The login name and password assigned to that remote system in your /etc/passwd file
- The phone number and speed of the modem attached to the local system

7.3 Running uucpsetup

The following sections describe how to use the uucpsetup script with the –a option to set up UUCP. With the uucpsetup script you can configure modems, incoming systems, outgoing systems, and the Poll file. See the uucpsetup(8) reference page for information about other options that are available for the uucpsetup script.

You can invoke the uucpsetup script either by choosing the UNIX-to-UNIX Copy Program (UUCP) option from the Setup Menu or by entering the following command:

```
# uucpsetup –a
```

The uucpsetup script prompts you for information required to configure modems, hardwired connections, and TCP/IP connections.

7.3.1 Configuring Modems, Hardwired Connections, and TCP/IP Connections

This section provides information on how to use the `uucpsetup` script to configure modems, hardwired connections, and TCP/IP connections. As you work through this section, the `uucpsetup` script modifies the following files with the answers you provide:

- `/etc/inittab`
- `/usr/lib/uucp/Devices`
- `/etc/inetd.conf`

Use the following procedure to configure modems, hardwired connections, and TCP/IP connections:

1. Enter the number that corresponds to the type of connection you are configuring. You can configure a maximum of 19 connections.

   From the Hardwired Connections Menu, choose the number that corresponds to the task you want to complete, as follows:

   - If you choose to configure modems, enter 1 and go to step 2.
   - If you choose to configure a direct hardwired connection, enter 2 and go to step 7.
   - If you choose to configure a TCP/IP connection, enter 3 and go to step 11.
   - If you choose to configure a particular system with a direct link, enter 4 and go to step 14.
   - If you have finished configuring the connections for your system, enter 5 and go to Section 7.3.2.

   The following example illustrates how to configure modems:

```
------------------------------
MODEM CONFIGURATION
------------------------------

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Hardwired connections MENU:

                1. Configure Modems
                2. Using a DIRECT hardwired line
                3. Using UUCP over TCP/IP
                4. Using a direct link to a remote system
                5. End of Modem(s) Configuration


Please enter the number of your selection (1/2/3/4/5): 1
```

2. Enter the name of the modem type.

   The uucpsetup script checks the /usr/lib/uucp/Dialers file and lists the modems that are available to be configured.

3. Enter the baud rate for the modem you are configuring.

   You must enter one of the baud rates listed (1200/2400/9600/Any).

4. Enter a device name.

   The uucpsetup script lists the available device names. Enter the last letter or number of the device name that you are configuring. For example, to configure the device tty01, enter 1.

```
********************************************
These are the available device names
tty00
tty01
********************************************

Select a device name.
Enter the last numeral/alphanumeric character
for the appropriate tty device name.
For example type '1' if selecting tty01:  1
Renaming /dev/tty01 to /dev/ttyd1
Warning: /dev/tty01 has been used
```

5. Indicate whether to add the device to your system.

   The uucpsetup script displays a summary of your responses for the device you are configuring, and then prompts you to add the device, to skip it, or to redisplay the summary. If the information is correct and you want to add the device, press Return.

6. Indicate whether you want to enter an entry for uugetty in the inittab file.

   If you want to enter an entry, select an ID for the process in the /etc/inittab file.

   The /etc/inittab file supplies the init program with instructions for creating and running initialization processes.

   The /etc/inittab file has the following format:

   Identifier:Runlevel:Action:Command

   The uucpsetup 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 already exist in the /etc/inittab file. If the ID that you assign already exists, the uucpsetup script prompts you to enter another ID.

Indicate whether the system will use the modem or direct line in shared mode.

For more information on the /etc/inittab file see the inittab(4) reference page.

After you enter the ID for the process in the /etc/inittab file, the uucpsetup script redisplays the Hardwired Connections Menu. The configuration of the modem is now complete. Go to step 1.

7. Enter the baud rate for the direct hardwired line you are configuring. (See step 3.)
8. Enter a device name. (See step 4.)
9. Indicate whether to add the device to your system. (See step 5.)
10. Select an ID for the process in the /etc/inittab file. (See step 6.)

    After you enter the ID for the process in the /etc/inittab file, the uucpsetup script redisplays the Hardwired Connections Menu. The configuration of the direct connection is not complete. Go to step 1.
11. Indicate whether you want to configure UUCP to be able to place outgoing calls over TCP/IP.

    To enable UUCP to place outgoing calls over TCP/IP, enter yes or press Return. 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.

    If you do not want to configure UUCP to be able to place outgoing calls over TCP/IP, enter no and go to the next step.
12. Indicate whether you want to configure UUCP to be able to accept incoming calls over TCP/IP.

    To enable UUCP to accept incoming calls over TCP/IP, enter yes or press Return. When you enable UUCP to accept incoming calls over TCP/IP, the /etc/inetd.conf file is modified and the following warning is displayed:

```
Warning: Restart inetd daemon.
uucp calls over TCP/IP won't be accepted unless this is done
Kill the inetd daemon and restart with the following command
/usr/sbin/inetd &
```

If you do not want to configure UUCP to be able to place incoming calls over TCP/IP, enter no.

The configuration of the TCP/IP connection is now complete. Go to step 1.

13. Indicate whether you want to configure another hardwired connection.

    To configure another hardwired connection, enter yes and go to step 1.

    If you do not want to configure another hardwired connection, enter no and go to Section 7.3.2.

14. Enter the name of the remote system.
15. Enter the baud rate for the direct hardwired line for the system you are configuring. (See step 3.)
16. Enter a device name. (See step 4.)
17. Indicate whether to add the device to your system. (See step 5.)
18. Select an ID for the process in the `/etc/inittab` file. (See step 6.)

    After you enter the ID for the process in the `/etc/inittab` file, the `uucpsetup` script redisplays the Hardwired Connections Menu.

    The configuration of a system using direct connection is now complete. Go to step 1.

7.3.2 Configuring Outgoing Systems

When you stop configuring modems, the `uucpsetup` script prompts you for information required to specify remote systems that your system can contact. Ask the administrator of the remote system for the login name and password that he or she has assigned for 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.

Some or all of the following files are modified when you configure outgoing systems:

- `/usr/lib/uucp/Dialcodes`
- `/usr/lib/uucp/Systems`

Use the following procedure to configure outgoing systems:

1. Enter the name of the remote system.

2. Enter the mode of the connection.

   Your choice and the subsequent questions depend on the devices you configured.

   If you select the MODEM option, go to step 3.

   If you select the TCP option, go to step 9.

   If you select the Direct option, go to step 14.

3. Enter the number that corresponds to the times when your system is allowed to call the remote host:

   ```
   1   Any time of any day
   2   Evenings (Mon-Fri 5pm - 8am, Sat & Sun all day )
   3   Nights   (Mon-Fri 11pm - 8am,  Sat all day  &  Sun until 5pm)
   4   Never
   ```

4. Choose an option for the baud rate.

   If you enter 1 (Select a BAUD RATE), the `uucpsetup` script prompts you for the speed. Enter a speed that corresponds to a device you configured in the `/usr/lib/uucp/Devices` file. If the device can be used at any speed choose option 2. The following example shows how to specify the baud rate 9600:

   ```
   Select the option for the BAUD RATE(transmission speed)


   There are two options:

        1. Select a BAUD RATE.

        2. Select this option if the device can be used at any speed.

   Please enter the number of your selection (1/2) [2]: 1
   Enter the speed: 9600
   ```

5. Select an option for entering the phone number of the remote system.

   If you enter 1 (The complete telephone number), the `uucpsetup` script prompts you for the phone number of the remote system.

   If you enter 2 (A dialing prefix and a telephone number), the `uucpsetup` script prompts you to enter a prefix to be 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.

After you enter the prefix, the `uucpsetup` 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. Enter the rest of the number.

6. Enter the login name and password for your system on the remote system.

   This information must match the information in the `/etc/passwd` file on the remote system.

7. Verify the information that you provided and, if it is correct, press Return.

8. Choose an *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, enter 1.
   - To specify no *expect-send* strings, enter 2.
   - To be prompted to enter *expect-send* strings, enter 3.

   After you choose an option, the system is added to the `/usr/lib/uucp/Systems` file and `sendmail` is restarted to update the mail configuration file. Go to step 19.

9. Enter the number that corresponds to the TCP/IP conversation protocol (`g`, `t`, `e`, or `f`).

10. Enter the number that corresponds to the times when your system is allowed to call the remote host. (See step 3.)

11. Enter the login name and password for your system on the remote system. (See step 6.)

12. Verify the information that you provided. If it is correct, press Return.

13. When asked to select an *expect-send* string, choose option 2 (Specify no expect-send strings). (See step 8.)

The system is added to the Systems file and sendmail is restarted to update the mail configuration file. Go to step 19.

14. Enter the number that corresponds to the times when your system is allowed to call the remote host. (See step 3.)

15. Select an option for the baud rate.

    Enter a speed that corresponds to the direct device you configured in the /usr/lib/uucp/Devices file.

16. Enter the login name and password for your system on the remote system. (See step 6.)

17. Verify the information that you provided, and if it is correct, press Return.

18. Choose an *expect-send* string. (See step 8.)

    After you choose any option, the system is added to the /usr/lib/uucp/Systems file and sendmail is restarted to update the mail configuration file.

19. Enter the name of another system to configure another outgoing system, or press Return to indicate that you are finished configuring outgoing systems.

7.3.3 Configuring Incoming Systems

When you are done configuring outgoing systems, the uucpsetup script prompts you for information required to specify the remote systems allowed to establish incoming UUCP connections.

Some or all of the following files are modified when configuring incoming systems:

- /etc/passwd
- /usr/lib/uucp/Permissions

Use the following procedure to configure incoming systems:

1. Enter the name of a remote system that is allowed to establish incoming UUCP connections.

2. Enter the name of your system.

   The default provided is the name that you assigned your system at installation.

3. Specify the login ID for the remote system.

   The new login ID is 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 with an uppercase

U added as a prefix. For example, if you specify `machine1` for incoming connections, the login ID, by convention, is `Umachine1`. However, you can select any login ID.

4. After you indicate the login ID, the `uucpsetup` script prompts you for a comment to add to the `/etc/passwd` file for this login ID. Adding a comment is optional.

5. The `uucpsetup` script invokes the `vipw` command. Press Return and, after viewing the entry in the `/etc/passwd` file, exit the editing session by entering `:wq`. Then supply a password for the new entry:

   ```
   Invoking 'vipw'.
   Hit 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/kj951Fq: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 passwd
   Changing password for Umachine1.
   New password:
   Retype new password:
   ```

6. Indicate whether you want to enter another login ID for this remote 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.

7. Indicate whether to use the `REQUEST` option.

   This option allows a remote system to ask for any queued work on the local system that is meant for that remote system. Including this option makes it easy for remote system users to transfer files to and execute commands on a local system. 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.

8. Indicate whether to use the SENDFILES option.

   This option permits 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. Security considerations at your site might require that you limit a remote system's access to the local system by using the default value (CALL) for this option.

9. Indicate whether you want to add any locations for the READ and WRITE options.

   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 uucico to be allowed to place files in that location.

10. Indicate whether you want to add any locations for the NOREAD and NOWRITE options.

    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.

11. Indicate which 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.

    You are prompted for each command separately.

    The default is the command rmail only.

12. Indicate whether you want to use the VALIDATE option.

    This option specifies that the calling remote system must use a specific ID and password. The use of any other ID from that remote system fails. Several systems can use the same ID.

    The VALIDATE option is meaningful only when the login ID and password are protected.

13. Indicate whether to use the CALLBACK option.

    This option indicates that the local system must contact the remote system before the remote system can transfer any files to the local system.

    If both systems use the CALLBACK option in their respective Permissions files, they will never be able to communicate with each other.

Once you have entered one system as an incoming system, the next time you invoke `uucpsetup` with the `-i` option the startup menu appears as follows:

```
------------------------------
INCOMING SYSTEMS CONFIGURATION
------------------------------

You have two choices

1. Specify a remote system name.

2. Specify OTHER; meaning you are specifying options for
all the other machines not specified in the "Permissions"
file but listed in the "Systems" file.

press RETURN if none of the two choices:

Please enter your selection (1,2,RETURN):
```

If you press Return without making a selection, the script terminates and the defaults for the options are not entered in the `Permissions` file.

7.3.4 Configuring the Poll File

To configure the `/usr/lib/uucp/Poll` file, invoke `uucpsetup` with the `-p` option and perform 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.

   When you press Return, the `Poll` file is updated.
4. To add another system to the `Poll` file, enter `y`. Otherwise, press Return to exit `uucpsetup`.

Setting Up the Network Time Protocol 8

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. NTP is implemented by the University of Toronto's `xntpd` daemon. The `/etc/ntp.conf` file is the configuration file for the daemon. For more information about NTP, see the *Network and Communications Overview.*

While setting up and configuring NTP, the following files are created or modified:

- `/etc/ntp.conf`
- `/etc/rc.config`

You can also choose to set your system time by the `rdate` command, which is explained in Section 18.3.

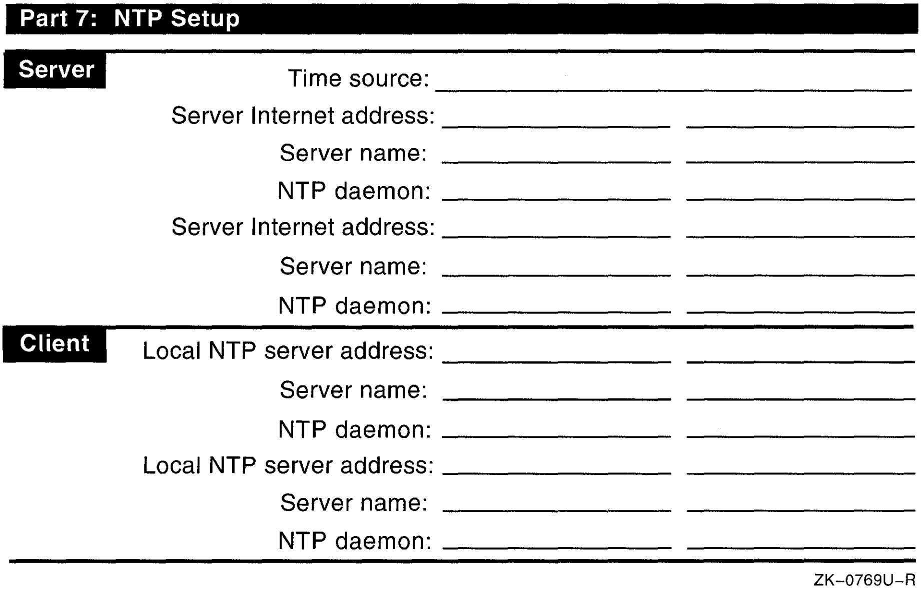
8.1 Gathering Information

Appendix A contains a worksheet that you can use to record the information that you need to complete the tasks in this book. Use Part 7 of the worksheet to record the information you gather as you work your way through this section. To obtain a copy of this worksheet, print the following PostScript file:

```
/usr/examples/network_configuration/worksheet.ps
```

Figure 8-1 shows Part 7 of the Configuration Worksheet.

Figure 8-1: Configuration Worksheet, Part 7



Part 7: NTP Setup

Server

Time source: ______________________________

Server Internet address: ______________ ______________

Server name: ______________ ______________

NTP daemon: ______________ ______________

Server Internet address: ______________ ______________

Server name: ______________ ______________

NTP daemon: ______________ ______________

Client

Local NTP server address: ______________ ______________

Server name: ______________ ______________

NTP daemon: ______________ ______________

Local NTP server address: ______________ ______________

Server name: ______________ ______________

NTP daemon: ______________ ______________

ZK-0769U-R

Gather the following information before setting up NTP:

- Whether your system is a local NTP server or an NTP client
- Your system's time source

  For local NTP servers, the time source will be one of the following:

  - Internet NTP servers

    If your system is connected to the Internet network, see Section 8.2 for information on obtaining a list of the NTP Internet servers and permission to use them.

    You must know the following about the Internet NTP servers:

    * The host name and IP address of the server
    * Whether they are running the `ntpd` or `xntpd` daemon

  - A local reference clock

    A local reference clock is a lightly loaded and highly available system that keeps good time. See Section 18.1.2 for information on setting up a local reference clock.

You must know the following about the local reference clock:

* The host name and IP address of the clock

* Whether it is running the ntpd or xntpd daemon

For NTP clients, the time sources are the systems specified as local NTP servers.

You must know the following about the local NTP servers:

- Their names and IP addresses.
- Whether they are running the ntpd or the xntpd daemon. Servers running the DEC OSF/1 operating system run the xntpd daemon.

8.2 Selecting Internet Servers

If you are setting up a local NTP server with Internet NTP servers as its time source, you must select the Internet servers you want to use. The list of possible Internet servers and information about their stratum level is available by means of anonymous File Transfer Protocol (FTP) from louie.udel.edu. In the following sample FTP session the list of NTP servers is copied from the system louie.udel.edu to the local host:

```
% ftp louie.udel.edu
220 louie.udel.edu FTP server (Version 4.108 Sun Feb 19 22:09:45 EST
1993) ready.
Name (louie.udel.edu:my_name): anonymous
Password (louie.udel.edu:anonymous): my_name
331 Guest login ok, send ident as password.
230 Guest login ok, access restrictions apply.
ftp> cd pub/ntp/doc
250 CWD command successful.
ftp> get clock.txt
200 PORT command successful.
150 Opening ASCII mode data connection for clock.txt (57002 bytes).
226 Transfer complete.
local: clock.txt remote: clock.txt
58409 bytes received in 14 seconds (4.2 Kbytes/s)
ftp> bye
221 Goodbye.
```

For security reasons, not all systems at a site can have anonymous FTP access.

You should select three systems from the list of Internet servers with which to synchronize the time on your local NTP servers. The systems that you select are called peers. Obtain permission from the contact person listed for the Internet server before specifying it as a peer for your local NTP servers.

If your network is not connected to the Internet network, you must select a system on your network to be the local reference clock. See Section 18.1.2 for information on setting up a local reference clock.

8.3 Running ntpsetup

With the `ntpsetup` script, you can configure all NTP clients; you can also configure local NTP servers if they use Internet NTP servers as their time source. However, local NTP servers that use a local reference clock as a time source should not use the `ntpsetup` script. For information on setting up local NTP servers that use a local reference clock, see Section 18.1.2.

Use the following procedure to set up NTP:

1. Invoke the `ntpsetup` script by choosing the Network Time Protocol (NTP) option from the Setup Menu or by entering the following command:

   ```
   # /usr/sbin/ntpsetup
   ```

   An explanation of `ntpsetup` is displayed on your screen.

2. Press Return following the script's explanation of what `ntpsetup` does.

3. Enter the names of the NTP servers for this system.

   For clients, enter the names of your site's three local NTP servers. For servers, enter the names of three Internet NTP servers. (See Section 8.2 for information on selecting Internet servers.)

   If you enter the name of a host that your system cannot find an address for in the local `/etc/hosts` database or through BIND or NIS, the `ntpsetup` script prompts you for its IP address:

   ```
   Hostname of NTP server [no default]: host1
           Looking up host host1
                Cannot find an address for "host1".
                To add "host1" to the /etc/hosts file, you must know
                "host1"'s Internet (IP) address.
      Would you like to add "host1" to the /etc/hosts
              file (y/n) [y]? Return
      What is host1's Internet (IP) address [no default] ?
              120.105.1.2
      Is 120.105.1.2 correct (y/n) [no default] ? y
      Is host1 running ntpd or xntpd (n/x) [x] ?Return
   Hostname of NTP server [no default]: host2
      Looking up host host2 ...found.
      Is host2 running ntpd or xntpd (n/x) [x] ? Return
   Hostname of NTP server [no default]: host3
      Looking up host host3 ...found.
      Is host3 running ntpd or xntpd (n/x) [x] ? Return
   Hostname of NTP server [no default]: Return
   ```

   The `ntpsetup` script displays the list of servers that you entered. If the list is correct, enter `c` to continue. If the list in incorrect or incomplete, enter `r` to redo it.

4. Press Return following the script's explanation that if any of your NTP servers are not on your subnet you must run either the `routed` or the `gated` daemon to access them.

For information on running the routed daemon, see Chapter 2.

5. Indicate whether you want to run the xntpd daemon with the –g option.

   The –g option allows xntpd to correct time differences of more than 1000 seconds between your system and that of your system's NTP servers that occur after the xntpd daemon is started. Initial time differences are corrected before the xntpd daemon is started by the ntpdate command, which is run at boot time by the /sbin/init.d/settime script. If your system is sensitive to security threats, do not use the –g option. If you do not use the –g option, time differences of more than 1000 seconds will cause the xntpd daemon to log a message to syslog and exit.

6. Indicate whether you want NTP to log only error messages and the initialization message.

   Although you can have only error messages and the initialization message logged to syslog (by entering y), Digital recommends that you configure NTP to log normal status messages as well. The status messages are logged infrequently, do not consume much disk space, and contain useful information.

   The ntpsetup script then displays a message similar to the following and exits:

   ```
   Configuring your system to run NTP...done.

   Starting the NTP daemon (xntpd)...
   Setting kernel timezone variable
   Setting the current time and date with ntpdate
   Fri Dec 06 11:48:15 EST 1992
   Network Time Service started

   To monitor NTP, type "/usr/bin/ntpq -p".
   ```

Setting Up Your Mail System 9

This chapter describes how to set up your DEC OSF/1 mail system by using the `mailsetup` script. When you use `mailsetup` to set up your mail system, the `/var/adm/sendmail/sendmail.cf` file is created or modified.

This chapter also provides information about the four mail utilities included in the DEC OSF/1 operating system and about the `sendmail` utility.

9.1 The Mail Systems

DEC OSF/1 operating system includes the following four mail utilities:

- The `mail`, `binmail` utility

  The `mail`, `binmail` utility, the default, is 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. Some of the user features of the `mail` utility are as follows:

  – Send, deliver, and read mail messages

  – Save messages in a mailbox (local delivery agent)

  – Write messages to a file

- The `mailx`, `Mail` utility

  The `mailx`, `Mail` utility is 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. Some of the user features of the `mailx` utility are as follows:

  – Send and receive mail messages

  – Save messages in a mailbox

  – Write messages to a file

  – Save messages in folders

  – Display debugging information

- The message handler utility (mh)

  The mh utility and its associated commands are included in the optional RAND Corporation Mail Handler subset. 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, the mh utility depends on the `mail` utility for delivery to a user's mailbox. Some of the features of the mh utility are as follows:

  - Send and receive mail messages
  - Save messages in a mailbox
  - Write messages to a file
  - Save messages in folders as individual files
  - Display debugging information
  - Provide graphical interface with the `dxmail` and `xmh` commands
  - Provide the Post Office Protocol (POP)

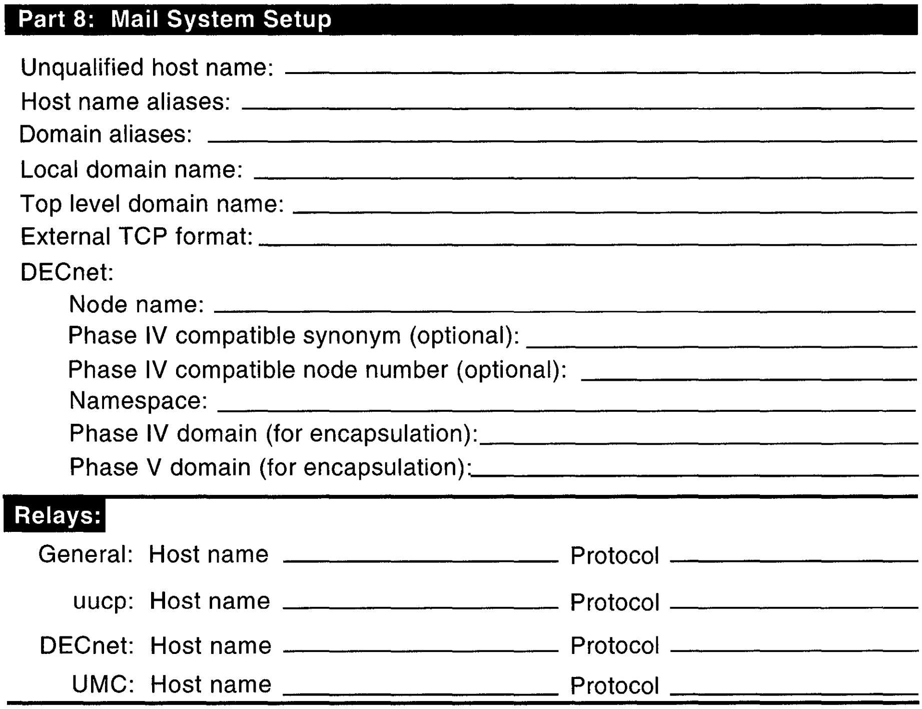
9.2 Gathering Information

Appendix A contains a worksheet that you can use to record the information that you need to complete the tasks in this book. Use Part 8 of the worksheet to record the information you gather as you work your way through this section. To obtain a copy of this worksheet, print the following PostScript file:

```
/usr/examples/network_configuration/worksheet.ps
```

Figure 9-1 shows Part 8 of the Configuration Worksheet.

Figure 9-1: Configuration Worksheet, Part 8



Part 8: Mail System Setup

Unqualified host name: ______

Host name aliases: ______

Domain aliases: ______

Local domain name: ______

Top level domain name: ______

External TCP format: ______

DECnet:

Node name: ______

Phase IV compatible synonym (optional): ______

Phase IV compatible node number (optional): ______

Namespace: ______

Phase IV domain (for encapsulation): ______

Phase V domain (for encapsulation): ______

Relays:

General: Host name ______ Protocol ______

uucp: Host name ______ Protocol ______

DECnet: Host name ______ Protocol ______

UMC: Host name ______ Protocol ______

ZK-0770U-R

Using `mailsetup`, you can either do a basic or advanced mail setup. If you do a basic mail setup, you only need to know the domain name and the name of your general relay host. The basic setup provides defaults for all other information. If you do an advanced mail setup, you will have to provide all of the following information:

- The unqualified host name and aliases

  The unqualified host name is the name of your machine without the domain extension. For example, a machine called `foo.dec.com` has an unqualified name of `foo`.

  Aliases are alternative names that other systems might use to direct mail to your host.

- The domain names

  If your host is part of a registered Berkeley Internet Name Domain (BIND) domain, you must know the name of the domain in which your host is registered. You must also know the names of other domains in which the host may be named.

You must also know the name of the top level domain for your site. The top level domain name is the name of the highest level of domain that all the systems at your site can reach without going through a gateway.

- The relays

  The `mailsetup` script enables you to set up relays for general purposes, UUCP, DECnet, and ULTRIX Mail Connection (UMC). To set up these relays, you must know the names of the systems that will perform these functions and the transport protocols that your system will use to send mail to the relays.

  The general relay is the host name of your general-purpose relay machine. Any mail that cannot be resolved locally is forwarded to this machine for processing. If the host is a TCP/IP host, this host name must be the fully qualified host name.

  The UUCP relay host enables you to receive mail for sites not directly reachable by this host. The DECnet relay will be the machine to which you pass DECnet mail.

  Without a UUCP relay, all UUCP mail to sites not known by this host will fail (returned to sender). Without a DECnet relay, all DECnet mail will fail.

  The UUCP and DECnet relays default to the name of the general relay host. Without a DECnet relay, all DECnet mail will fail. This defaults to the general relay.

  You must know which protocol the host will use to send mail to the general relay. The transport protocol can be TCP, UUCP, or DECnet (if installed).

  If you specify UUCP as the transport protocol, you must be sure it is set up.

- DECnet information

  The DECnet information you need depends on the following:

  - Whether DECnet is installed on the machine
  - Whether you plan to process DECnet mail locally or send it to a relay for processing

  If you plan to process DECnet mail locally, you need the following information:

  - The DECnet Phase V node name
  - The DECnet Phase IV compatible synonym of the machine
  - The DECnet Phase IV compatible node number for the machine

  If you plan to send DECnet mail to a relay for processing, you must know the fully qualified name of the DECnet relay and the transport protocol for the DECnet relay, as explained previously.

9.3 User Configurable Mail Locking

DEC OSF/1 enables you to configure the locking style. To do this, use the `/usr/sbin/rcmgr` command to set `MAILLOCKING` in the `/etc/rc.config` file. For more information, see Section 19.1.4.

9.4 Running mailsetup

To use the `mailsetup` script to set up your mail system, log in as `root` and complete the following steps:

1. Invoke the `mailsetup` script by choosing the Mail option from the Setup Menu or by issuing the following command:

   ```
   #/usr/sbin/mailsetup
   ```

2. If you are not running BIND, the `mailsetup` script asks if you want to run it. If you answer yes, the `mailsetup` script calls the `bindsetup` script. For more information on `bindsetup`, see Chapter 4.
3. The `mailsetup` script then asks whether you want to do a quick setup.

   If you answer yes, the `mailsetup` script prompts you for the following information:

   - The name of the general-purpose relay
   - If you want to modify the list of aliases and users that are considered local
   - If you want to complete the mail setup

   To do an advanced mail setup, answer no. The `mailsetup` script prompts you for the information you collected on the worksheet.
4. When you finish providing the information, the `mailsetup` script asks you if you want to complete the configuration. If you answer yes, the script moves the new `sendmail.cf` file to the system space, saves the old `sendmail.cf` file, and restarts `sendmail`. If you answer no, the script moves the new `sendmail.cf` file to `/var/adm/sendmail/sendmail.cf.tmp` and exits.

For more information on `mailsetup`, see the `mailsetup`(8) reference page.

Setting Up the Simple Network Management Protocol Agent 10

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

> The DEC OSF/1 software supports the POLYCENTER Common Agent implementation of the SNMP Agent. It does not implement the NMS software.

During the setup and configuration process the following files are created or modified:

- `/etc/eca/internet_mom.conf`
- `/etc/eca/snmp_pe.conf`

10.1 Editing the snmp_pe.conf File

The `/etc/eca/snmp_pe.conf` file is the configuration file for the `snmp_pe` daemon. You must edit it to add information about the communities and trap communities you want configured on your system, and to indicate whether or not to disable authentication failure traps.

The following default entry in the `/etc/eca/snmp_pe.conf` file allows any Network Management Station (NMS) to monitor your system:

```
community public  0.0.0.0 readonly
```

To configure specific communities, remove this entry from your file and replace it with your own entry. Community entries in the `/etc/eca/snmp_pe.conf` file have the following format:

community *community_name NMS_IP_address* **community-type**

Trap community entries have the following format:

trap *trap_community_name NMS_IP_address*

To disable authentication failure traps, you must add the following entry:

```
no_auth_traps
```

The following is a sample /etc/eca/snmp_pe.conf file with the test1, test2, and test3 communities configured:

```
#
# SNMP network management agent configuration database
#
community        test1  128.45.10.100 readonly
community        test1  16.45.7.110   readonly
community        test2  130.160.4.22  readonly
community        test3    0.0.0.0     readwrite
#
trap             test1  128.45.10.100
```

The test1 community can be monitored by the NMS whose IP address is 128.45.10.100 or by the one whose IP address is 16.45.7.110. The test2 community can be monitored by NMS 130.160.4.22 only. The test3 community can be monitored and managed by any NMS within the test3 community.

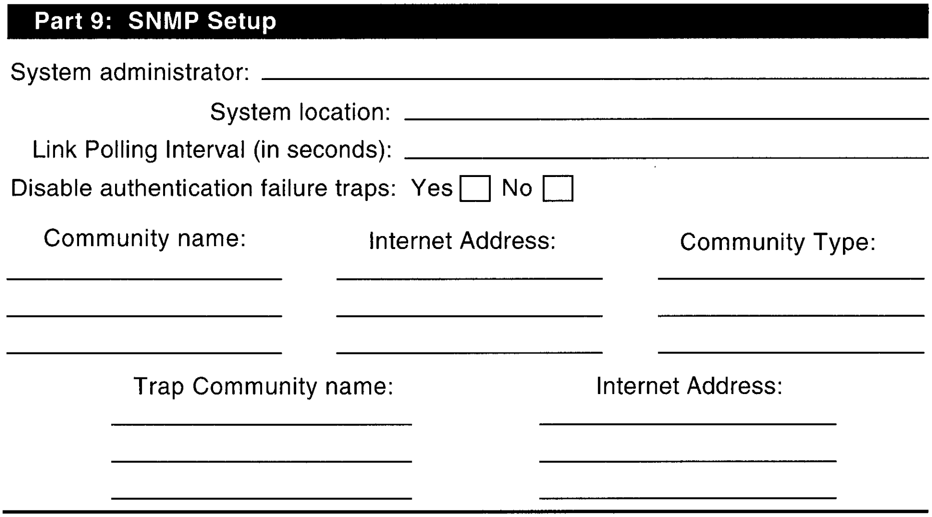
10.2 Gathering Information

Appendix A contains a worksheet that you can use to record the information that you need to complete the tasks in this book. Use Part 9 of the worksheet to record the information you gather as you work your way through this section. To obtain a copy of this worksheet, print the following PostScript file:

```
/usr/examples/network_configuration/worksheet.ps
```

Figure 10-1 shows Part 9 of the Configuration Worksheet.

Figure 10-1: Configuration Worksheet, Part 9



Part 9: SNMP Setup

System administrator: ____________________

System location: ____________________

Link Polling Interval (in seconds): ____________________

Disable authentication failure traps: Yes □ No □

Community name:	Internet Address:	Community Type:

Trap Community name:	Internet Address:

ZK-0771U-R

Gather the following information before setting up SNMP:

- The name of the system administrator of the local system.
- The physical location of the system.
- The Link Polling Interval (in seconds) to be used by the `/usr/sbin/internet_mom` daemon to monitor the state of each TCP/IP interface on your system.
- The community name or names.

  The community name is used by the SNMP protocol to authenticate requests from an NMS.

  You can configure multiple communities.
- The IP address or addresses that you want associated with a community name.

  Specify the IP addresses of the NMSs that you want to associate with the community name. You can specify multiple NMSs for the same community name. You can also specify the IP address `0.0.0.0` with any community name. This allows any NMS within the specified community to monitor the system.
- The community type defines the type of access that the community has. The community types can be one of the following:

  - `readonly`
  - `readwrite`
  - `writeonly`
  - `none`

- The trap community name or names.

  The trap community name is used by the SNMP protocol to send SNMP traps to NMSs that are listening for SNMP traps.

  You can configure multiple trap communities.

- The IP address or addresses you want associated with the trap communities.

- Whether you want authentication failure traps enabled or disabled.

10.3 Running snmpsetup

Use the following procedure to set up the SNMP Agent:

1. Invoke the `snmpsetup` script by choosing the Simple Network Management Protocol (SNMP) option from the Setup Menu or by entering the following command:

   ```
   # /usr/sbin/snmpsetup
   ```

   An explanation of `snmpsetup` is displayed on your screen.

2. Press Return following the script's explanation of `snmpsetup`.

3. Enter the name of the system administrator, the physical location of the system, and the default link polling interval.

4. Enter a community name.

   If you press Return without entering a community name, the script indicates that community information is required for SNMP to work on your system and offers to configure a community named `public` with an associated IP address of `0.0.0.0`. If you want SNMP to work on your system, you must configure the community `public` if you did not configure any other communities.

   If you do not want SNMP to work on your system, answer no when `snmpsetup` offers to configure the community `public`.

5. Enter an IP address to associate with the community named in step 4.

   Enter the IP address of an NMS that is allowed to monitor the system. If you specify an IP address of 0.0.0.0, any NMS within the named community can monitor your system.

   If you want multiple NMSs to monitor the same community, answer yes when snmpsetup asks if you want to configure another community. Enter the same community name but specify the IP address of another NMS. For example:

```
Enter community name [RETURN when done] : test
Enter IP address associated with community
      test [0.0.0.0]? 128.16.45.10
   •
   •
   •
Do you wish to add another community [n]? y
Enter community name [RETURN when done] : test
Enter IP address associated with community
      test [0.0.0.0]? 128.16.45.15
```

6. Enter a community type for the community.
7. Indicate whether you want to add another community.

   After you complete entering communities, enter a trap community name to enable sending SNMP traps to specific NMSs.
8. Enter an IP address to associate with the trap community named in step 7, if any.
9. Indicate whether you want to add another trap community.
10. Indicate whether you want to disable authentication failure traps. After you have completed answering all of the questions, the snmpsetup script indicates what system files it is updating.
11. Indicate whether you want to restart the POLYCENTER Common Agent (the snmp_pe, mold, fddi_mom, internet_mom and trn_mom daemons).

    To effect the changes that snmpsetup has made, restart the POLYCENTER Common Agent daemons, by answering yes.

    If you answer no, use the following commands to stop and restart the daemons manually after snmpsetup exits and returns you to the system prompt (#):

    ```
    # /sbin/init.d/common_agent stop
    # /sbin/init.d/common_agent start
    ```

    Some of the variables in the Internet Protocol (IP) routing table and the Exterior Gateway Protocol (EGP) group are obtained from the gated daemon, if it is running on the system. If the gated daemon is not running prior to starting the Common Agent daemons, the default values are used for these variables.

10.4 Extending the SNMP Agent

The POLYCENTER Common Agent is extensible. If you have the optional POLYCENTER Common Agent Developer's Toolkit layered product, you can add Managed Object Modules (MOMs) for managing objects other than the MIB-II objects managed by the `internet_mom` daemon, the FDDI objects managed by the `fddi_mom` daemon, and the *IEE 802.5 Token Ring MIB* objects managed by the `trn_mom` daemon. For more information, see the POLYCENTER Common Agent Developer's Toolkit documentation.

10.5 More Information

For more information about SNMP, see the `fddi_mom`(8), `internet_mom`(8), `mold`(8), `snmp_pe`(8), `snmpsetup`(8), and `trn_mom`(8) reference pages.

The following Requests for Comments (RFCs) contain information about SNMP and the Management Information Base (MIB):

- *Structure and Identification of Management Information for TCP/IP-Based Internets* (RFC 1155)
- *Management Information Base for Network Management of TCP/IP-Based Internets* (RFC 1156)
- *A Simple Network Management Protocol (SNMP)* (RFC 1157)
- *Management Information Base for Network Management of TCP/IP–Based Internets: MIB-II* (RFC 1213)
- *Conventions for Defining Traps for Use With the SNMP* (RFC 1215)
- *IEE 802.5 Token Ring MIB* (RFC 1231)

Part 2: Alternative Setup Methods

Chapters 11 through 20 describe how to manually configure the following:

- Transmission Control Protocol/Internet Protocol (TCP/IP) Network (Chapter 11)
- Serial Line Internet Protocol (SLIP) (Chapter 12)
- Local Area Transport (LAT) (Chapter 13)
- Berkeley Internet Name Domain (BIND) Service (Chapter 14)
- Network Information Service (NIS) (Chapter 15)
- Network File System (NFS) (Chapter 16)
- UNIX-to-UNIX Copy Program (UUCP) (Chapter 17)
- Network Time Protocol (NTP) (Chapter 18)
- Mail (Chapter 19)
- Simple Network Management Protocol (SNMP) Agent (Chapter 20)

Manually Setting Up the Network 11

This chapter describes how to manually set up the network, which includes the following tasks:

- Configuring the network interfaces
- Optionally, enabling the following network daemons:
  - rwhod
  - routed
  - gated
  - writesrv
- Optionally, setting up a router
- Optionally, setting up static routes
- Adding hosts to the /etc/hosts file
- Optionally, adding hosts to the /etc/hosts.equiv file
- Optionally, adding network names to the /etc/networks file
- Starting the network

11.1 Configuring Network Interfaces

Use the following procedure to configure the network interfaces on your system:

1. Check to see if the host name is set for your system by entering the following command:

   ```
   # /sbin/hostname
   ```

   If your system does not have a host name, set it by modifying HOSTNAME in the /etc/rc.config file by using the rcmgr command. For example, to set your host name to zzanny, you would enter the following command:

```
# /usr/sbin/rcmgr set HOSTNAME zzanny
```

2. Set the number of network interfaces you want to configure on your system by modifying NUM_NETCONFIG in the /etc/rc.config file, using the rcmgr command. For example, if you wanted to configure two interfaces on your system, you would enter the following command:

```
# /usr/sbin/rcmgr set NUM_NETCONFIG 2
```

3. The maximum number of network devices you can have in your hardware configuration is system dependent. Set the value of MAX_NETDEVS in the /etc/rc.config file to this maximum, using the rcmgr command. For example, if your hardware can support a maximum of 24 network devices, you would enter the following command:

```
# /usr/sbin/rcmgr set MAX_NETDEVS 24
```

Note

The maximum number of network devices currently supported by netsetup is 24.

4. There is one NETDEV_*n* entry in the /etc/rc.config file for each network device you want to configure on your system. Set the name of the network device you want to configure by modifying the NETDEV_*n* entry in the /etc/rc.config file, using the rcmgr command. This command has the following syntax:

   /usr/sbin/rcmgr set NETDEV_*n device*

   The value of *n* can be from 0 to 1 less than the value of MAX_NETDEVS. The *device* parameter specifies the name of the network device on your system (for example ln0, fza0).

5. There is a pair of NETDEV_*n* and IFCONFIG_*n* entries in the /etc/rc.config file for each network device that you configure on your system.

   The IFCONFIG_*n* entry defines the ifconfig command parameters for the corresponding NETDEV_*n* device. Set the ifconfig command parameters for the corresponding NETDEV_*n* device by modifying IFCONFIG_*n* in the /etc/rc.config file, using the rcmgr command. Enclose the parameters in double quotation marks and separate each field with a space. The syntax of this command varies depending on the type of network device you are configuring:

   - If you are configuring an Ethernet device, the syntax is as follows:

     /usr/sbin/rcmgr set IFCONFIG_*n "address netmask mask* *parameters"*

The parameters are as follows:

n

Is a number from 0 to 1 less than the value of `MAX_NETDEVS`. For example, set `IFCONFIG_0` to the `ifconfig` parameters for device `NETDEV_0`.

address

Is the IP address of the `NETDEV_`*n* device. The *address* parameter can alternatively be the host name.

netmask

Is a keyword indicating that the following string identifies the network mask.

mask

Is the network mask.

parameters

Are optional additional `ifconfig` parameters. For example, you might want to specify no trailers or a different broadcast address. Additional parameters that you specify are dependent on your network configuration. If you have no additional parameters, omit them from the `rcmgr` command that sets `IFCONFIG_`*n*.

- If you are configuring a SLIP device, the syntax is as follows:

  /usr/sbin/rcmgr set IFCONFIG_*n "address rem_address netmask mask parameters"*

  The parameters are the same as the Ethernet device with the addition of the *rem_address* parameter. This parameter is defined as follows:

  rem_address

  Is the IP address of the remote SLIP interface.

- If you are configuring a Token Ring device, the syntax is as follows:

  /usr/sbin/rcmgr set IFCONFIG_*n "address netmask mask speed number parameters"*

  The parameters are the same as the Ethernet device with the addition of the *speed* and *number* parameters. These parameters are defined as follows:

speed
: Is a keyword indicating that the following number defines the speed of the Token Ring adapter.

number
: Is the speed of the Token Ring adapter. The speed can be either 4Mb or 16Mb. The default speed is 16Mb.

See `ifconfig`(8) for more information.

If your system has more than one network interface, repeat steps 4 and 5 for the other network interfaces on your system.

6. For SLIP interfaces, there is a `SLIPTTY_`*n* in the `/etc/rc.config` file for each `NETDEV_`*n* SLIP device entry that you configure on your system.

   The `SLIPTTY_`*n* entry defines the `slattach` command parameters for the corresponding `NETDEV_`*n* and `IFCONFIG_`*n* entries. Set the `slattach` command parameters by modifying `SLIPTTY_`*n* in the `/etc/rc.config` file using the `rcmgr` command. This command has the following syntax:

   /usr/sbin/rcmgr set SLIPTTY_*n "[flags] ttyname [baudrate]"*

   The parameters are as follows:

   n
   : Is a number from 0 to 1 less than the value `MAX_NETDEVS`. For example, set `SLIPTTY_2` to the `slattach` parameters for SLIP device `NETDEV_2`.

   flags
   : Are optional `slattach` parameters. For example, you might want to enable TCP header compression. If you do not want any flags, omit them from the `rcmgr` command that sets `SLIPTTY_`*n* .

   ttyname
   : Is the name of any valid terminal device in the `/dev` directory. This can be either the full path name (for example, `/dev/tty01`) or the name in the `/dev` directory (for example, `tty01`).

   baudrate
   : Is the speed of the connection. The default speed is 9600 baud.

   See `slattach`(8) for more information.

7. Add an entry in the `/etc/hosts` file for your host, using the procedure in Section 11.5.

   If your system has more than one network interface, each interface might or might not have a name. Add an entry to the `/etc/hosts` file for each interface on your system that has a name, using the procedure in Section 11.5.

8. Optionally, enable network daemons, set up a router, add static routes, or add entries to network configuration files by using the procedures in Section 11.2 to Section 11.7.
9. Start the network, using the procedure in Section 11.8.

   See Section 11.9 for more network configuration information.

11.2 Enabling and Disabling Network Daemons

This section explains how to enable and disable the following network daemons:

- rwhod
- routed
- gated
- writesrv

You can choose to run either the routed or the gated daemon; however, you cannot run both.

11.2.1 Running the rwhod Daemon

The rwho daemon (rwhod) maintains the database used by the rwho and ruptime commands. Running rwhod is optional; however, it must be running to use these commands.

11.2.1.1 Starting and Enabling the rwho Daemon

To start the rwhod daemon, perform the following steps:

1. If the network is started, check to see if the rwhod daemon is running by issuing the following command:

   ```
   # /bin/ps ax | grep rwhod
   ```

2. If the network is started and the rwhod daemon is not running, enter the following command to start the rwhod daemon in the background:

   ```
   # /usr/sbin/rwhod
   ```

If you enable the rwhod daemon, it is started automatically by the /sbin/init.d/rwho script each time the network is restarted or the system is rebooted. Use the rcmgr command to modify the entry for the rwhod daemon in the /etc/rc.config file:

```
# /usr/sbin/rcmgr set RWHOD yes
```

11.2.1.2 Stopping and Disabling the rwho Daemon

To stop the rwhod daemon, perform the following steps:

1. Check to see if the rwhod daemon is running by issuing the following command:

   ```
   # /bin/ps ax | grep rwhod
   ```

2. If the rwhod daemon is running, kill the process by issuing a /bin/kill command with the process ID (PID) for the daemon obtained from the /bin/ps command.

If you disable the rwhod daemon, it is not started automatically by the /sbin/init.d/rwho script each time you restart the network or reboot the system. Use the following rcmgr command to disable the rwhod daemon in the /etc/rc.config file:

```
# /usr/sbin/rcmgr set RWHOD no
```

For more information, see rwhod(8).

11.2.2 Running the routed Daemon

The routed daemon automatically updates the internal routing tables in your host. It does this by using the Routing Information Protocol (RIP). Running the routed daemon is optional.

Note

> You cannot run both the routed daemon and the gated daemon on your system.

11.2.2.1 Starting and Enabling the routed Daemon

To start the routed daemon on your system, perform the following steps:

1. If the network is started, check to see whether the routed daemon is running by issuing the following command:

   ```
   # /bin/ps ax | grep routed
   ```

2. If you want to add static routes, use the procedure in Section 11.4.
3. If the network is started and the routed daemon is not running, you can start the routed daemon with or without flags.

   To start the routed daemon without flags, enter the following

command:

```
# /usr/sbin/routed
```

To start the routed daemon with flags, include the flags in the command line, separating each flag with a space. For example, the following command starts the routed daemon with the -s flag, which causes the routed daemon to supply RIP information even if it is not functioning as an Internet router:

```
# /usr/sbin/routed -s
```

For more information, see routed(8).

If you enable the routed daemon, it is started automatically by the /sbin/init.d/route script each time the network is restarted or the system is rebooted. To enable the routed daemon, perform the following steps:

1. Use the following rcmgr command to enable the routed daemon:

   ```
   # /usr/sbin/rcmgr set ROUTED yes
   ```

2. Check to see if the routed daemon flags are set in /etc/rc.config, using the following rcmgr command:

   ```
   # /usr/sbin/rcmgr get ROUTED_FLAGS
   ```

   If flags are set and you do not want any routed daemon flags, reset the flags in /etc/rc.config, using the following rcmgr command:

   ```
   # /usr/sbin/rcmgr set ROUTED_FLAGS ''
   ```

   If you want to change the routed daemon flags, reset the flags in /etc/rc.config, using the rcmgr command. Enclose the flags in double quotation marks and separate each flag with a space.

   See routed(8) for more information.

11.2.2.2 Stopping and Disabling the routed Daemon

To stop the routed daemon, perform the following steps:

1. Check to see if the routed daemon is running by issuing the following command:

   ```
   # /bin/ps ax | grep routed
   ```

2. If the routed daemon is running, kill the process by issuing a /bin/kill command with the process ID (PID) for the daemon obtained from the /bin/ps command.

If you disable the routed daemon, it is not started automatically by the /sbin/init.d/route script each time you restart the network or reboot the system. To disable the routed daemon, perform the following steps:

1. Use the following rcmgr command to disable the routed daemon:

   ```
   # /usr/sbin/rcmgr set ROUTED no
   ```

2. If the routed daemon flags are set, you can reset the flags in the /etc/rc.config file by issuing the following rcmgr command:

   ```
   # /usr/sbin/rcmgr set ROUTED_FLAGS "
   ```

   See routed(8) for more information.

11.2.3 Running the gated Daemon

The gated daemon automatically updates the internal routing tables in your host. It can do this using multiple routing protocols. Running the gated daemon is optional.

Note

> You cannot run both the routed daemon and the gated daemon on your system.

11.2.3.1 Starting and Enabling the gated Daemon

To start the gated daemon, perform the following steps:

1. If the network is started, check to see whether the gated daemon is running by issuing the following command:

   ```
   # /bin/ps ax | grep gated
   ```

2. The /etc/gated.conf file contains configuration information that is read by the gated daemon. If the /etc/gated.conf file does not exist, set it up in the format specified in gated.conf(4).

   If the /etc/gated.conf file exists, you can modify it if needed.

   If the gated daemon is running when you modify the /etc/gated.conf file, the gated daemon detects the changes and they take effect immediately. Otherwise, the changes take effect when you manually start the gated daemon.

3. If you want to add static routes, use the procedure in Section 11.4.

4. If the network is started and the gated daemon is not running, you can start it with or without flags.

To start the gated daemon without flags, issue the following command:

```
# /usr/sbin/gated
```

To start the gated daemon with flags, include the flags in the command line, separating each flag with a space. For example, the following command starts the gated daemon with the -r flag, which causes the gated daemon to log all routing changes:

```
# /usr/sbin/gated -r
```

For more information, see gated(8).

If you enable the gated daemon, it is started automatically by the /sbin/init.d/gateway script each time the network is restarted or the system is rebooted. To enable the gated daemon, perform the following steps:

1. Use the following rcmgr command to enable the gated daemon:

   ```
   # /usr/sbin/rcmgr set GATED yes
   ```

2. Check to see if the gated daemon flags are set in the /etc/rc.config file, using the following rcmgr command:

   ```
   # /usr/sbin/rcmgr get GATED_FLAGS
   ```

   If flags are set and you do not want any gated daemon flags, reset the flags in the /etc/rc.config file, using the following rcmgr command:

   ```
   # /usr/sbin/rcmgr set GATED_FLAGS "
   ```

   If you want to change the gated daemon flags, reset the flags in the /etc/rc.config file, using the rcmgr command. Enclose the flags in double quotation marks and separate each flag with a space.

   See gated(8) for more information.

11.2.3.2 Stopping and Disabling the gated Daemon

To stop the gated daemon, perform the following steps:

1. Check to see if the gated daemon is running by issuing the following command:

   ```
   # /bin/ps ax | grep gated
   ```

2. If the gated daemon is running, kill the process by issuing a /bin/kill command with the process ID (PID) for the daemon obtained from the /bin/ps command.

If you disable the gated daemon, it is not started automatically by the /sbin/init.d/gateway script each time you restart the network or reboot the system. To disable the gated daemon, perform the following steps:

1. Use the following rcmgr command to disable the gated daemon:

   ```
   # /usr/sbin/rcmgr set GATED no
   ```

2. If gated daemon flags are set, you can reset the flags in the /etc/rc.config file, using the following rcmgr command:

   ```
   # /usr/sbin/rcmgr set GATED_FLAGS "
   ```

   See gated(8) for more information.

11.2.4 Running the writesrv Daemon

The writesrv daemon receives remote write command requests. Running writesrv is optional; however, it must be running to use the following options with the write command:

- –h
- –q
- –r

11.2.4.1 Starting and Enabling the writesrv Daemon

To start the writesrv daemon, perform the following steps:

1. If the network is started, check to see if the writesrv daemon is running by issuing the following command:

   ```
   # /bin/ps ax | grep writesrv
   ```

2. If the network is started and the writesrv daemon is not running, enter the following command to start the writesrv daemon in the background:

   ```
   # /usr/sbin/writesrv
   ```

If you enable the writesrv daemon, it is started automatically by the /sbin/init.d/write script each time the network is restarted or the system is rebooted. Use the rcmgr command to modify the entry for the writesrv daemon in the /etc/rc.config file:

```
# /usr/sbin/rcmgr set WRITESRV yes
```

11.2.4.2 Stopping and Disabling the writesrv Daemon

To stop the writesrv daemon, perform the following steps:

1. Check to see if the `writesrv` daemon is running by issuing the following command:

   ```
   # /bin/ps ax | grep writesrv
   ```

2. If the `writesrv` daemon is running, kill the process by issuing a `/bin/kill` command with the process ID (PID) for the daemon obtained from the `/bin/ps` command.

If you disable the `writesrv` daemon, it is not started automatically by the `/sbin/init.d/write` script each time you restart the network or reboot the system. Use the following `rcmgr` command to disable the `writesrv` daemon in the `/etc/rc.config` file:

```
# /usr/sbin/rcmgr set WRITESRV no
```

For more information, see `writesrv`(8).

11.3 Setting Up an IP Router

An IP router (also called a gateway) connects two or more local area networks (LANs). A router allows data to be transferred between systems on the networks to which it is connected.

To set up an IP router, perform the following steps:

1. Configure the network interfaces on your system, using the procedure in Section 11.1.
2. Set the global variables `ipforwarding` and `ipgateway` in the running kernel by issuing the following command:

   ```
   # /usr/sbin/iprsetup -s
   ```

3. Set the value of ROUTER in the `/etc/rc.config` file to indicate that your system is set up as an IP router by issuing the following `rcmgr` command:

   ```
   # /usr/sbin/rcmgr set ROUTER yes
   ```

11.4 Adding Static Routes

If the network is started, you can use the `/usr/sbin/route` command to add a route immediately. The format of the `/usr/sbin/route` command is as follows:

/usr/sbin/route add { **-net** | **-host** } *destination* [**-interface**] *gateway*

`-net`
: Specifies the destination is a network.

`-host`
: Specifies the destination is a host.

`destination`
: Specifies the name or IP address of the destination host or network. Specifies the keyword ''default'' when adding a default gateway.

`-interface`
: Optionally, specifies that the route is through an interface.

`gateway`
: Specifies the name or IP address of the gateway or interface.

See `route`(8) for more information.

A route that you add with the `/usr/sbin/route` command is in effect until you reboot the system, restart the network, or issue a `/usr/sbin/route flush` command. If you want the route to be established each time you reboot the system or restart the network, you must add an entry to the `/etc/routes` file. When the network restarts, the `/sbin/init.d/route` script runs, and executes a `/usr/sbin/route add` command for each entry in the `/etc/routes` file.

The format for an entry in the `/etc/routes` file is described in `routes`(4).

11.5 Adding Hosts to the /etc/hosts File

The `/etc/hosts` file contains the names and addresses of other hosts on your network to which you want to connect. If your network currently uses or will be set up to use either the Network Information Service (NIS, formerly YP) or the Berkeley Internet Name Domain (BIND) service to distribute host information, you do not need a complete listing of all hosts on your network in your `/etc/hosts` file. However, you should include the names and addresses of hosts that are (or will be) designated as servers for those services.

The format of an entry in the `/etc/hosts` file is as follows:

IP_address host1 [*alias_1 alias_2 alias_n*] [*# comment*]

The following is a sample /etc/hosts file:

```
# @(#n)hosts    1.0     (DEC OSF/1)
#
# Description:  The hosts file associates host names with
#               IP addresses.
#
# Syntax: nnn.nnn.nnn.nnn hostname.domain.name [alias_1,...,\
#         alias_n] [#comments]
#
# nnn.nnn.nnn.nnn       The IP address of the host.
# hostname.domain.name  The fully qualified host name, including
#                       the domain name.
# alias_n               Other names or abbreviations for this host.
# #comments             Text following the comment character (#)
#                       is ignored.
#
127.0.0.1 localhost
120.105.5.1 host1.cities.dec.com h1
120.105.5.2 host2.cities.dec.com h2
120.105.5.3 host3.cities.dec.com h3     #BIND server
120.105.5.4 host4.cities.dec.com h4     #BIND server
120.105.5.5 host5.cities.dec.com h5
```

See hosts(4) for more information.

Note

If you change the IP address or host name in the /etc/hosts file, associated with any network interfaces you have configured, you might need to change the IP address or host name on the corresponding IFCONFIG_*n* line in the /etc/rc.config file.

11.6 Adding Hosts to the /etc/hosts.equiv File

Users on a host specified in the /etc/hosts.equiv file can log in to your system without password verification, if they have a valid account on your system. You can restrict access to your system without password verification to specific users by specifying a host and a user name in the /etc/hosts.equiv file.

The format of the /etc/hosts.equiv file is as follows:

host1
host2 user1

The following is a sample /etc/hosts.equiv file:

```
# @(#)hosts.equiv    1.0     (DEC OSF/1)
#
# Description:  The hosts.equiv file contains a list of
#               trusted hosts.
#
# Warning:  Listing hosts in this file can compromise system
#           security.  Include host names and user names in
#           this file with caution.
#
# Syntax: host1 [username]
#
# host1         Name of a host considered trusted by the
#               local system.
# [username]    Individual user who can log in to the local
#               system without supplying a password.
#
host1
host2 diane
host2 charlotte
host2 kate
```

In the preceding example, all users with accounts on host1 can log in to the local system without specifying a password. Users diane, charlotte, and kate on host2 can log in to the local system without specifying a password. For more information on the /etc/hosts.equiv file, see hosts.equiv(4).

11.7 Adding Network Names to the /etc/networks File

The /etc/networks file allows the netstat command to translate network numbers into network names. If you do not enter network names into the /etc/networks file, the netstat command displays network numbers instead of network names. Entries in the /etc/networks file have the following format:

name number [*alias_1 ... alias_n*] [*# comment*]

The following is a sample /etc/networks file:

```
# @(#)networks   1.0   (DEC OSF/1)
#
# Description:   The networks file lists the known networks in the
#          Internet.
#
# Syntax: network_name network_number [ alias_1 ... alias_n ] [ #comment ]
#
# network_name   Name of the network supplied by the network
#          administrator.
# network_number  Network number assigned to the network by the NIC.
# alias_n    One or more other names or abbreviations for this network.
# #comments  Text following the comment character (#) is ignored.
#
```

```
loop         127   loopback
ethernet1    98    doconet
ethernet2    100   devonet
```

See networks(4) for more information.

Note

> If your network is running NIS, the networks database is distributed. If the networks database is distributed, you must edit the master copy of the networks database in the /var/yp/src directory on the NIS master servers and remake the maps for it. For information about updating and remaking NIS maps, see the *Network Administration and Problem Solving* manual.

11.8 Starting the Network

After you finish setting up the network, you can start the network by using the rcinet command.

If the network is stopped, start the network by entering the following command:

```
# /usr/sbin/rcinet start
```

If the network is already started, warn the network users on your system in advance that the network on your system is being restarted. File systems that were not mounted using the /etc/fstab file or the automount command must be unmounted with the unmount command (see mount(8)). You must remount these file systems after the network is restarted.

Restart the network by entering the following command:

```
# /usr/sbin/rcinet restart
```

See rcinet(8) for more information.

Alternatively, you can start the network by rebooting the system with the following command:

```
# shutdown –r now
```

The –r option specifies an automatic reboot.

11.9 More Information

For more information, see gated(8), ifconfig(8), netstat(1), slattach(8), routes(4), and routed(8).

For information about IP addresses, subnetworks, network classes, and routing, see the *Network and Communications Overview.*

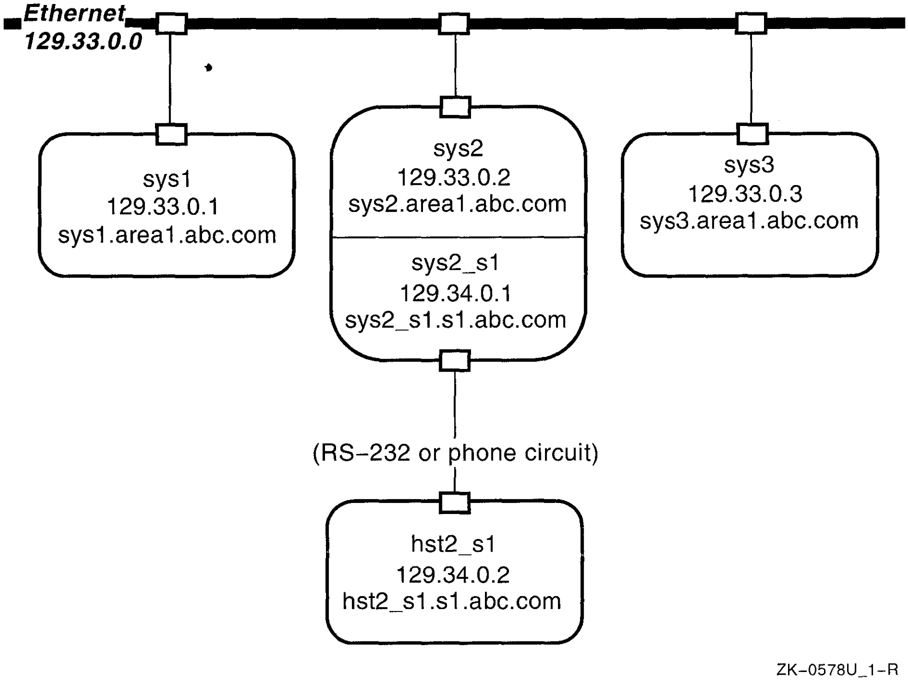
For information about network management see the *Network Administration and Problem Solving* manual.

Setting Up the Serial Line Internet Protocol 12

The Serial Line Internet Protocol (SLIP) is a protocol used to run IP over serial lines, including RS-232 cables connecting two systems and telephone circuits. Unlike Ethernet, a serial line provides a point-to-point connection between only two hosts. Like Ethernet, TCP/IP commands (such as `rlogin`, `ftp`, and `ping`) can be run over the SLIP connection.

Figure 12-1 provides an example of a network with a serial line.

Figure 12-1: Using SLIP to Connect a Remote System to a Network



This example shows a remote host (`hst2_s1`) connected to a network by a SLIP connection to host `sys2`. This host (`sys2`) is different from the others in the example because it uses two network adapters – one for Ethernet and one for the serial line connection.

TCP/IP requires a unique Internet address for each adapter used. In this example, a new subnetwork is created for the SLIP connection (129.34.0.0). The serial line adapters used to create this network are given appropriate addresses for this subnetwork (129.34.0.1 for sys2 and 129.34.0.2 for hst2_sl). Additionally, a name is assigned for the serial interface on sys2, sys2_sl.

If properly configured, hst2_sl is able to communicate with host sys2 and the other hosts in the 129.33.0.0 network.

12.1 Managing Routing

You can use either the routed or the gated daemon to manage routing, if you are not using the SLIP connection solely to communicate between the two systems making the connection. In Figure 12-1, all the systems must run either the routed or gated daemon for hst2_sl to be able to communicate with sys1 or sys3 and for them to be able to communicate with hst2_sl. In the example, sys2 acts as a router for the end systems (sys1, sys3, and hst2_sl). Make certain that the routed daemon running on the systems at each end of the SLIP connection do not run in quiet mode (the -q flag).

If you plan to use a system as an IP router (sys2 in Figure 12-1), it must be configured to allow the forwarding of IP packets. For more information on setting a system up as an IP router, see Section 11.3 and iprsetup(8).

You should restart the routed or gated process if either was running prior configuring the SLIP interface with the ifconfig command. This ensures that the SLIP interface is recognized by the routed or gated daemon.

12.2 Using SLIP Physical Connections

You can use SLIP to connect systems either directly (using a null modem) or over telephone lines using modems.

If you connect the systems directly, use an RS-232 cable to connect the serial ports on the two hosts. The cable used must be a null modem cable, such as Digital BC22D-*xx*, (where *xx* varies depending on the length of the cable).

You can use this method for hosts in close proximity to each other. The maximum length of this type of connection is defined by the RS-232 standard.

If the systems are not in close proximity to each other, you can connect them using telephone line and modems. To use this kind of connection, attach a modem to a serial port on both hosts so that the two hosts can establish a serial connection between them. You can use an RS-232 cable connected to the serial port on the host. This cable must be a straight-through cable such as Digital BC22E-*xx* or BC22F-*xx* and the modems must be set to 8 bit no parity.

12.3 Setting Up SLIP

To configure your system to use SLIP, perform the following steps:

1. Add the SLIP pseudodevices to the host's kernel.

   By default, an entry for SLIP exists in the system configuration file, `/sys/conf/` *HOSTNAME*, (where *HOSTNAME* is the name of your system). Therefore, there is no need to modify this file unless this entry has been modified or deleted.

   The entry appears as follows:

   ```
   pseudo-device     sl     2
   ```

   This entry provides the host with up to two SLIP interfaces (sl0 and sl1).

   For more information, see the section on configuring the kernel in the *System Administration* manual.

2. Add entries to the `/etc/hosts` file for the network adapters at both ends of the SLIP network.

   For example, using the hosts shown in Figure 12-1, you would make the following entries on both `sys2` and `hst2`:

   ```
   129.34.0.1       sys2_sl  sys2_sl.sl.abc.com
   129.34.0.2       hst2_sl  hst2_sl.sl.abc.com
   ```

3. Configure the SLIP interfaces by using the `ifconfig` command. For `sys2` in Figure 12-1, you could issue the following command:

   ```
   # ifconfig sl0 129.34.0.1 129.34.0.2 netmask 255.255.0.0
   ```

   On `hst2`, you could issue the following command:

   ```
   # ifconfig sl0 129.34.0.2 129.34.0.1 netmask 255.255.0.0
   ```

   In these examples, the first argument is the name of the SLIP pseudodevice, the second argument is the local address of the SLIP interface, the third argument is the address of the SLIP interface on the remote host, and the remaining arguments specify the network mask.

   For more information, see the `ifconfig`(8) reference page.

4. Attach a serial line to a SLIP interface by using the `slattach` command.

   You use the `slattach` command to select the serial line that will be attached to the SLIP interface. The `slattach` command is also used to enable or disable the SLIP options. For more information on specifying options, see the `slattach`(8) reference page.

   When you use the `slattach` command, you do not specify the SLIP interface to be used (sl0, sl1,...). Instead, the first configured SLIP interface (one for which you issued an ifconfig command) that is not already attached is used. You also specify the baud rate for the serial connection with the slattach command. If you do not specify a rate, it it uses the default rate of 9600 baud. The following is an example of using the slattach command:

```
# slattach tty00 9600
```

In this example, tty00 attaches to a SLIP interface and sets the baud rate to 9600. The connection will use the options that were previously set. (When you boot the system, no SLIP options are enabled.)

The following is another example of the slattach command:

```
# slattach +c -i tty00
```

In this example, tty00 attaches to a SLIP interface running at 9600 baud (the default speed). TCP header compression is enabled and ICMP traffic suppression is disabled. (Disabling an option has no effect, if it was previously disabled.)

For more information on the slattach command, refer to the slattach(8) reference page.

After completing these steps, the SLIP network is available as long as the physical connection is ready and the slattach command is running. The ps command can be used to ensure the slattach command is running. If at any time, the slattach command exits (due to a system or network error), the command can be executed again to reestablish the SLIP network.

The physical connection is always ready if a direct connection is being used. For phone connections, the connection is readied by manually dialing the modem on the local system to connect to the modem on the remote system. Once the remote modem answers, the data/talk button (or equivalent) should be pressed to allow the modem on the local host to assume control of the connection. Depending on the type of modem used, the connection might take a few seconds while the modems negotiate speeds, protocols, and other session parameters before it is ready for use. The connection should stay up until one side hangs up due to some error or a user intentionally disconnects by pressing the data/talk button.

12.4 Stopping and Restarting SLIP

You stop the SLIP network by using the kill(1) command to stop the running slattach process that has attached a serial line to SLIP. You can restart the SLIP network by reissuing the slattach command, which readies the physical connection. When disabling a SLIP network that runs over a phone connection, press the data/talk button on the modem to hang up the line.

Manually Setting Up the Local Area Transport 13

This chapter explains how to set up the Local Area Transport (LAT) on your system manually.

To run LAT on your system, you must also configure LAT in your system's kernel (see Section 3.2).

Optionally, you can customize your LAT setup. For information on customizing your LAT setup, see Section 3.4. The customization section includes information on general customization, setting up printers, host-initiated connections, the LAT/Telnet gateway, and creating your own service.

13.1 Setting Up LAT

To set up LAT on your system manually, you must first configure your kernel for LAT (see Section 3.2). Then, log in as superuser and perform the following steps:

1. Create the LAT device special files by running the `/dev/MAKEDEV` script and specify its LAT option.

   This script creates one LAT device special file for each LAT terminal device. Each time you invoke the `MAKEDEV` script, it creates 16 LAT device special files. The `MAKEDEV` script also creates the LAT device (`/dev/lat`) that can be cloned, if it has not already been created. The `MAKEDEV` script requires 16 contiguous LAT device special files to be available. (Specifying `lat38` requires only 12 contiguous LAT device special files to be available.)

   For example, the following commands create 32 device special files for LAT devices:

   ```
   #  cd /dev
   # MAKEDEV lat0
   # MAKEDEV lat1
   ```

   The option range is 0 to 38.

   Record the device special file information displayed by `MAKEDEV`. The actual special file names vary depending on how many other terminal devices are already configured.

The following is a sample MAKEDEV display:

```
MAKEDEV: special file(s) for lat1:
tty16 tty17 tty18 tty19 tty1a tty1b tty1c tty1d
tty1e tty1f tty1g tty1h tty1i tty1j tty1k tty1l
```

Check to see that the files were created correctly by issuing the `/sbin/ls -l` command and by specifing the `/dev/tty`*WX* file. (Where *W* is a number from 0 to 9 and *X* is an alphanumeric from 0 to 9, lowercase a to z, or uppercase A to Z.)

2. Edit the `/etc/inittab` file to include entries for the LAT device special files you created.

   For example:

   ```
   lat16:3:respawn:/usr/sbin/getty tty16
   lat17:3:respawn:/usr/sbin/getty tty17
   lat18:3:respawn:/usr/sbin/lattelnet tty18 lattelnet18
   ```

   For detailed information, see the `inittab`(8) reference page.

3. Use the following command to spawn the processes for the LAT device special files that you added to the `/etc/inittab` file:

   ```
   # /sbin/init q
   ```

4. Create the STREAMS special file required by LAT by issuing the following command:

   ```
   # /usr/sbin/strsetup -i
   ```

   After creating the STREAMS special file, check to make certain it was successful by issuing the following command:

   ```
   # /sbin/ls -l /dev/streams/kinfo
   ```

5. Enable LAT automatic startup and shutdown by using the following command:

   ```
   # /usr/sbin/rcmgr set LAT_SETUP 1
   ```

   When LAT automatic startup and shutdown is enabled, the `/sbin/init.d/lat` file automatically starts LAT upon reaching run level 3 and automatically stops LAT when exiting run level 3.

6. Start LAT by issuing the following command:

   ```
   # /usr/sbin/latcp -s
   ```

13.2 More Information

For more information on LAT, see Chapter 3 and the following reference pages: `init`(8), `inittab`(4), `latsetup`(8), `latcp`(8), MAKEDEV(8), `netsetup`(8), `rcmgr`(8), and `strsetup`(8).

Manually Setting Up the Berkeley Internet Name Domain Service 14

This chapter describes how to set up the Berkeley Internet Name Domain (BIND) service on your network manually. Setting up a BIND domain includes configuring the following:

- Primary server
- Secondary servers
- Caching servers
- Slave servers
- Clients

14.1 Setting Up the Primary Server

There can be only one primary server in a BIND domain. Use the following procedure to set up a BIND primary server:

1. Create the `/etc/resolv.conf` file.

   The `/etc/resolv.conf` file contains the domain name and the Internet (IP) address for the local host. Format the `/etc/resolv.conf` file as follows, substituting your domain name for `cities.dec.com`:

   ```
   # @(#)resolv.conf
   #
   # Description:  The resolv.conf file lists name-value pairs that
   #               provide information to the BIND resolver.
   #
   # Syntax:       domain  <domainname>
   #                      and
   #               nameserver  <address>
   #
   # Caution:  White space entered after the domain name is not
   #           ignored; it is interpreted as part of the domain name.
   #
   # domain <domainname>       local domain name
   # nameserver <address>      Internet address of a name server
   #                           that the resolver should query
   #
   domain          cities.dec.com
   nameserver      127.0.0.1
   ```

2. Create the database files by using the following procedure:

   a. Copy into or create in the `/etc/namedb/src` directory a file called `hosts`. The `hosts` file should have the following format:

   ```
   127.0.0.1 localhost
   120.105.1.20 host1.cities.dec.com h1        #BIND server
   120.105.1.142 host2 h2
   120.105.1.1 host3 h3                        #BIND server
   120.105.1.13 host4
   120.105.2.23 host5 h5
   ```

   The first field is the IP address. The second field is the host name and the third field contains any aliases for the host name. The fourth field, beginning with the number sign (#), contains comments. The third and fourth fields are optional.

   b. Run the `make hosts` command from within the `/etc/namedb` directory. Enter the following commands:

   ```
   # cd /etc/namedb
   # make hosts
   ```

   The `make hosts` command creates the `/etc/namedb/hosts.db` and `/etc/namedb/hosts.rev` files.

   ### Note

   Any host names with a domain name different from that for which you are creating the database are ignored. For example, if you create the `hosts` database for the domain `cities.dec.com` and you have a host name `fizzle.nac.dec.com` in the file, `fizzle.nac.dec.com` is ignored. Also, the first host name that the `make hosts` command encounters that has either no domain name or the default domain name becomes the primary name of the machine. All other names are considered aliases, or CNAMES. For example, for the following entry, the `make hosts` command considers `host2` the primary name of the system and `h2` an alias:

   ```
   120.105.1.20 host2 h2
   ```

3. Create the `/etc/namedb/named.ca` file.

The /etc/namedb/named.ca file must read as follows:

```
;
; BIND data file for initial cache data for root domain servers.
;
.               99999999        IN      NS      ns.nic.ddn.mil.
.               99999999        IN      NS      ns.nasa.gov.
.               99999999        IN      NS      terp.umd.edu.
.               99999999        IN      NS      a.isi.edu.
.               99999999        IN      NS      aos.brl.mil.
.               99999999        IN      NS      gunter-adam.af.mil.
.               99999999        IN      NS      c.nyser.net.
ns.nic.ddn.mil. 99999999        IN      A       192.67.67.53
ns.nasa.gov.    99999999        IN      A       128.102.16.10 ; BIND
                99999999        IN      A       192.52.195.10
a.is.edu.       99999999        IN      A       26.3.0.103
                99999999        IN      A       128.9.0.107
aos.brl.mil.    99999999        IN      A       128.20.1.2    ; BIND
                99999999        IN      A       192.5.25.82
gunter-adam.af.mil. 99999999    IN      A       26.1.0.13
c.nyser.net.    99999999        IN      A       192.33.4.12   ; BIND
terp.umd.edu.   99999999        IN      A       128.8.10.90   ; BIND
```

4. Create the /etc/namedb/named.local file.

   The /etc/namedb/named.local file must contain the following information and be formatted as shown in the following example. Replace host1.cities.dec.com with your host and domain name.

```
;
; BIND data file for local loopback interface.
;
@ IN SOA host1.cities.dec.com. postmaster.host1.cities.dec.com. (
                        1       ; Serial
                        3600    ; Refresh
                        300     ; Retry
                        3600000 ; Expire
                        3600 )  ; Minimum
    IN      NS      host1.cities.dec.com.
1   IN      PTR     localhost.
localhost.  IN      A       127.0.0.1
```

5. Create the boot file.

   The following is a sample named.boot file for a primary server. Replace cities.dec.com with your domain name and 120.105 with your network number:

```
# @(#)named.boot
#
# Description:  The named.boot file is required to boot a BIND
#               name server.
#
# Syntax:   directory   <directory_name>
#           ;[comment]
#           type                domain          source host/file
#
# <directory_name>   Location where domain data files are stored.
# ;[comment]         Text following the ';' character is ignored.
# type               Specifies primary, secondary, or forwarder
#                     server.
# domain              Name of the BIND domain.
# source host         IP address of the server distributing the
#                     database listed under 'file'; not applicable
#                     for primary servers.
# file                Name of database being distributed by
#                     'source host'.
#
directory        /etc/namedb
;
; type           domain                         source host/file
primary          cities.dec.com                    hosts.db
primary          120.105.in-addr.arpa              hosts.rev
;
;
primary          0.0.127.in-addr.arpa             named.local
;
; load the cache data last
cache            .                                named.ca
```

The default directory in which the database files are stored is `/etc/namedb`. You can store database files in any directory; however, if you place them in a directory other than the default directory, you should change the `/etc/namedb` in the boot file to the name of the directory you have chosen.

6. Edit the `/etc/rc.config` file by using the `/usr/sbin/rcmgr` utility. The syntax for the `/usr/sbin/rcmgr` command is as follows:

   /usr/sbin/rcmgr set *variable value*

   Enter the following commands to edit the `/etc/rc.config` file and add the required information:

   ```
   # /usr/sbin/rcmgr set BIND_CONF YES
   # /usr/sbin/rcmgr set BIND_SERVERTYPE PRIMARY
   # /usr/sbin/rcmgr set BIND_SERVERARGS "-b \
     /etc/namedb/named.boot"
   ```

7. Edit the `/etc/hosts` file with the fully qualified BIND name of the host.

   To run BIND, your system's host name must include the BIND domain name. The fully qualified BIND host name consists of the local host name plus the BIND domain name, separated by dots. For example, the fully qualified BIND host name for a system whose local host name is `host1` and whose BIND domain name is `cities.dec.com` is `host1.cities.dec.com`.

   See the `hosts`(4) reference page for more information.

8. Edit the `/etc/rc.config` file by using the `/usr/sbin/rcmgr` utility. The syntax for the `/usr/sbin/rcmgr` command is as follows:

/usr/sbin/rcmgr set *variable value*

Enter the following command to edit the `/etc/rc.config` file and add the required information:

```
# /usr/sbin/rcmgr set HOSTNAME host1.cities.dec.com
```

Replace `host1.cities.dec.com` with your system's fully qualified BIND name.

9. Set the new host name with the `/sbin/hostname` command.

   For example, to set the host name to `host1.cities.dec.com` for a system that was previously known locally as `host1`, enter the following command:

   ```
   # /sbin/hostname host1.cities.dec.com
   ```

10. Start the `named` daemon by issuing the following command:

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

14.2 Setting Up a Secondary Server

Use the following procedure to set up a BIND secondary server:

1. Create the `/etc/resolv.conf` file. See step 1 in Section 14.1.
2. Create the `/etc/namedb/named.ca` file. See step 3 in Section 14.1.
3. Create the `/etc/namedb/named.local` file. See step 4 in Section 14.1.
4. Create the boot file.

   A boot file for a secondary server should have the format shown in the following example. Replace `cities.dec.com` with your domain name, `120.105` with your network number, and `120.105.4.5` with the IP address of your domain's BIND primary server:

   ```
   # @(#)named.boot
   #
   # Description:  The named.boot file is required to boot a BIND
   #               name server.
   #
   # Syntax:   directory   <directory_name>
   #           ;[comment]
   #           type                domain          source host/file
   #
   # <directory_name>    Location where domain data files are stored.
   # ;[comment]          text following the ';' character is ignored.
   # type                Specifies primary, secondary, or forwarder
   #                     server.
   # domain              Name of the BIND domain.
   # source host         IP address of the server distributing the
   #                     database listed under 'file'; not applicable
   ```

```
#                    for primary servers.
# file               Name of database being distributed by
#                    'source host'.
#
directory       /etc/namedb
;
; type          domain                     source host/file
secondary       cities.dec.com             120.105.4.5   hosts.db
secondary       120.105.in-addr.arpa       120.105.4.5   hosts.rev
;
;
primary         0.0.127.in-addr.arpa    named.local
;
; load the cache data last
cache           .                       named.ca
```

The following entry indicates that this host serves itself its own local host information:

```
primary         0.0.127.in-addr.arpa    named.local
```

The default directory in which the database files are stored is /etc/namedb. You can store them in any directory; however, if you place them in a directory other than the default directory you should change the /etc/namedb at the top of the boot file to the name of the directory you have chosen.

5. Edit the /etc/rc.config file by using the /usr/sbin/rcmgr utility. The syntax for the /usr/sbin/rcmgr command is as follows:

   /usr/sbin/rcmgr set *variable value*

   Enter the following commands to edit the /etc/rc.config file and add the required information:

   ```
   # /usr/sbin/rcmgr set BIND_CONF YES
   # /usr/sbin/rcmgr set BIND_SERVERTYPE SECONDARY
   # /usr/sbin/rcmgr set BIND_SERVERARGS "-b \
       /etc/namedb/named.boot"
   ```

6. Edit the /etc/hosts file to add the fully qualified BIND name of the host.

   In order to run BIND, your system's host name must include the BIND domain name. The fully qualified BIND host name consists of the local host name plus the BIND domain name, separated by dots. For example, the fully qualified BIND host name for a system whose local host name is host2 and whose BIND domain name is cities.dec.com is host2.cities.dec.com.

   See the hosts(4) reference page for more information.

7. Edit the /etc/rc.config file by using the /usr/sbin/rcmgr utility. The syntax for the /usr/sbin/rcmgr command is as follows:

/usr/sbin/rcmgr set *variable value*

Enter the following command to edit the /etc/rc.config file and add the required information:

```
# /usr/sbin/rcmgr set HOSTNAME host2.cities.dec.com
```

Replace host2.cities.dec.com with your system's fully qualified BIND name.

8. Set the new host name with the /sbin/hostname command.

   For example, to set the host name to host2.cities.dec.com for a system that was previously known locally as host2, enter the following command:

   ```
   # /sbin/hostname host2.cities.dec.com
   ```

9. Start the named daemon by issuing the following command:

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

14.3 Setting Up a Caching Server

Use the following procedure to set up a BIND caching server:

1. Create the /etc/resolv.conf file. See step 1 in Section 14.1.
2. Create the /etc/namedb/named.ca file. See step 3 in Section 14.1.
3. Create the /etc/namedb/named.local file. See step 4 in Section 14.1.
4. Create the boot file.

   The following is a sample named.boot file for a caching server. Replace information that is appropriate for a caching server in your domain:

   ```
   # @(#)named.boot
   #
   # Description:  The named.boot file is required to boot a BIND
   #               name server.
   #
   # Syntax:   directory   <directory_name>
   #           ;[comment]
   #           type                domain          source host/file
   #
   # <directory_name>   Location where domain data files are stored.
   # ;[comment]         Text following the ';' character is ignored.
   # type               Specifies primary, secondary, or forwarder
   #                    server.
   # domain             Name of the BIND domain.
   # source host        IP address of the server distributing the
   #                    database listed under 'file'; not applicable
   #                    for primary servers.
   # file               Name of database being distributed by
   ```

```
#                      'source host'.
#
directory       /etc/namedb
;
; type          domain                  source host/file
primary         0.0.127.in-addr.arpa    named.local
;
; load the cache data last
cache           .                       named.ca
```

The default directory in which the database files are stored is `/etc/namedb`. You can store them in any directory; however, if you place them in a directory other than the default directory you should change the `/etc/namedb` entry at the top of the boot file to the name of the directory you have chosen.

5. Edit the `/etc/rc.config` file by using the `/usr/sbin/rcmgr` utility. The syntax for the `/usr/sbin/rcmgr` command is as follows:

   /usr/sbin/rcmgr set *variable value*

   Enter the following commands to edit the `/etc/rc.config` file and add the required information:

   ```
   # /usr/sbin/rcmgr set BIND_CONF YES
   # /usr/sbin/rcmgr set BIND_SERVERTYPE CACHING
   # /usr/sbin/rcmgr set BIND_SERVERARGS "-b \
       /etc/namedb/named.boot"
   ```

6. Edit the `/etc/hosts` file with the fully qualified BIND name of the host.

   In order to run BIND, your system's host name must include the BIND domain name. The fully qualified BIND host name consists of the local host name plus the BIND domain name, separated by dots. For example, the fully qualified BIND host name for a system whose local host name is `host3` and whose BIND domain name is `cities.dec.com` is `host3.cities.dec.com`.

   See the `hosts`(4) reference page for more information.

7. Edit the `/etc/rc.config` file by using the `/usr/sbin/rcmgr` utility. The syntax for the `/usr/sbin/rcmgr` command is as follows:

   /usr/sbin/rcmgr set *variable value*

   Enter the following command to edit the `/etc/rc.config` file and add the required information:

```
# /usr/sbin/rcmgr set HOSTNAME host3.cities.dec.com
```

Replace `host3.cities.dec.com` with your system's fully qualified BIND name.

8. Set the new host name with the `/sbin/hostname` command.

   For example, to set the host name to `host3.cities.dec.com` for a system that was previously known locally as `host3`, enter the following command:

   ```
   # /sbin/hostname host3.cities.dec.com
   ```

9. Start the `named` daemon by issuing the following command:

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

14.4 Setting Up a Slave Server

Use the following procedure to set up a BIND slave server:

1. Create the `/etc/resolv.conf` file. See step 1 in Section 14.1.
2. Create the `/etc/namedb/named.local` file. See step 4 in Section 14.1.
3. Create the boot file. The following is a sample `named.boot` file for a slave server. Replace `120.105.4.5` with the IP address of the BIND primary server in your domain:

   ```
   ;
   ; BIND data file to boot a slave name server.
   ;
   ; directory where all the data files are stored
   directory       /etc/namedb
   ;
   ; type          domain                  source host/file
   primary         0.0.127.in-addr.arpa    named.local
   ;
   slave
   forwarders      120.105.4.5
   ```

   The default directory in which the database files are stored is `/etc/namedb`. You can store them in any directory; however, if you place them in a directory other than the default directory you should change the `/etc/namedb` entry at the top of the boot file to the name of the directory you have chosen.

4. Edit the `/etc/rc.config` file by using the `/usr/sbin/rcmgr` utility. The syntax for the `/usr/sbin/rcmgr` command is as follows:

   /usr/sbin/rcmgr set *variable value*

Enter the following commands to edit the /etc/rc.config file and add the required information:

```
# /usr/sbin/rcmgr set BIND_CONF YES
# /usr/sbin/rcmgr set BIND_SERVERTYPE SLAVE
# /usr/sbin/rcmgr set BIND_SERVERARGS "-b \
    /etc/namedb/named.boot"
```

5. Edit the /etc/hosts file and add the fully qualified BIND name of the host.

   In order to run BIND, your system's host name must include the BIND domain name. The fully qualified BIND host name consists of the local host name plus the BIND domain name, separated by dots. For example, the fully qualified BIND host name for a system whose local host name is host4 and whose BIND domain name is cities.dec.com is host4.cities.dec.com.

   See the hosts(4) reference page for more information.

6. Edit the /etc/rc.config file by using the /usr/sbin/rcmgr utility. The syntax for the /usr/sbin/rcmgr command is as follows:

   /usr/sbin/rcmgr set *variable value*

   Enter the following command to edit the /etc/rc.config file and add the required information:

   ```
   # /usr/sbin/rcmgr set HOSTNAME host4.cities.dec.com
   ```

   Replace host4.cities.dec.com with your system's fully qualified BIND name.

7. Set the new host name with the /sbin/hostname command.

   For example, to set the host name to host4.cities.dec.com for a system that was previously known locally as host4, enter the following command:

   ```
   # /sbin/hostname host4.cities.dec.com
   ```

8. Start the named daemon by issuing the following command:

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

14.5 Setting Up a Client

Use the following procedure to set up a BIND client:

1. Create the /etc/resolv.conf file.

   The /etc/resolv.conf file for a client contains the domain name and IP addresses of up to three servers for the domain. These name servers are the systems that the local host can query to resolve host information. Format the /etc/resolv.conf file as follows, replacing your domain name for cities.dec.com and the IP addresses of your name servers for 120.105.4.5, 120.105.4.13, and 120.105.5.160:

```
;
; BIND data file
;
domain           cities.dec.com
nameserver       120.105.4.5
nameserver       120.105.4.13
nameserver       120.105.5.160
```

See step 1 in Section 14.4.

2. Edit the /etc/rc.config file by using the /usr/sbin/rcmgr utility. The syntax for the /usr/sbin/rcmgr command is as follows:

   /usr/sbin/rcmgr set *variable value*

   Enter the following commands to edit the /etc/rc.config file and add the required information:

```
# /usr/sbin/rcmgr set BIND_CONF YES
# /usr/sbin/rcmgr set BIND_SERVERTYPE CLIENT
```

14.6 Post-Setup Procedures

After you configure BIND on your system, you must restart the sendmail process, and edit the /etc/svc.conf file.

14.6.1 Restarting the sendmail Process

To kill and restart the sendmail process, enter the following command:

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

14.6.2 Editing the svc.conf File

The /etc/svc.conf file is the database service selection configuration file that your system references to determine what distributed database lookup services are running on your system, which databases are being served by them, and in what order to query them. After configuring BIND, you must edit the /etc/svc.conf file to tell your system that you want BIND servers queried for host name and address information. For information on editing the /etc/svc.conf file, see Appendix B.

Manually Setting Up the Network Information Service 15

This chapter describes how to set up the Network Information Service (NIS) manually. Setting up NIS includes configuring the following:

- Master server
- Slave servers
- Clients

Note

> You must have the Additional Networking Services subset installed to create an NIS master or slave server.

15.1 Setting Up the Master Server

There can be only one master server in an NIS domain. To set up a master server, log in as superuser and perform the following steps:

1. Create the `/var/yp/src/mail.aliases` file.

   The `mail.aliases` file defines networkwide mail aliases. Creating this file is optional. However, if you want to define and distribute mail aliases on your network, you must create it. If you choose not to create a `mail.aliases` file, while the NIS maps are being built, an informational message displays on the screen that the `mail.aliases` file could not be found.

   For information on defining mail aliases, see the `aliases`(4) reference page.

2. Create the `/var/yp/src/netgroup` file.

   The `netgroup` file defines networkwide groups and is used for permission checking when doing remote mounts, remote logins, and remote shells. Creating this file is optional. However, if you want to define and distribute network group information on your network, you must create it. If you choose not to create a `netgroup` file, while the NIS maps are being built, an informational message displays on the screen that the `netgroup` file could not be found.

   For information on defining network groups, see `netgroup`(4).

3. Copy into the /var/yp/src directory the local /etc files that you intend to make into NIS maps for distribution and be sure that all of the information in the files is up to date.

   When the default set of NIS maps is created, the following file names are recognized in the /var/yp/src directory: aliases, group, hosts, mail.aliases, netgroup, networks, passwd, protocols, rpc, and services. If you do not want to distribute one of the default maps, do not copy the local /etc file for it into the /var/yp/src directory. If a file is absent from the /var/yp/src directory, while the NIS maps are being built, an informational message displays on the screen that the file could not be found.

4. Remove the entry for root from the passwd file after you copy it into the /var/yp/src directory.

5. Copy automount maps, or other site-specific maps, into the /var/yp/src directory. For information on creating automount maps, see Appendix D. For information on creating other site-specific maps, see the *Network Administration and Problem Solving* manual.

6. Create the domain directory by entering the following command, replacing test_domain with the name that you have chosen for your domain:

   ```
   # mkdir /var/yp/test_domain
   ```

7. Edit the /var/yp/Makefile file, if necessary.

   If you are using the NIS master server to serve the auto.master map, the auto.home map, or both, you must remove the number sign (#) from the beginning of the following lines:

   ```
   #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) - $(YPDBDIR)/$(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) - $(YPDBDIR)/$(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 number sign (#) at the beginning of the following lines:

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

8. Create the ypservers map.

   The ypservers map is a list of all of the domain's slave servers. To create this map, enter the following command, replacing test_domain with the name that you have chosen for your domain and replacing slave1, slave2, to slave*n* with the names of the slave servers:

   ```
   # /var/yp/makedbm - /var/yp/test_domain/ypservers
   slave1
   slave2
   slaven
   Ctrl/d
   ```

   Be sure to include on this list all of the slave servers in your domain. If you add a slave server to your domain, you must remake this list.

9. Create the NIS maps.

   To create the NIS maps that are distributed throughout the domain, edit the /var/yp/Makefile file to add the domain name you have chosen.

For example, if your domain name is `test_domain`, edit the `/var/yp/Makefile` file in the following way:

```
   •
   •
   •
#
#  ***** DOM must be filled in with the domain name !!
#
DOM=test_domain
   •
   •
   •
```

Then change to the `/var/yp/`*`nis_domain`* directory (`test_domain` in this example) and run the following command:

```
# cd test_domain
# make -f /var/yp/Makefile NOPUSH="Y"
```

10. Decide whether to run the `yppasswdd` daemon.

    The `yppasswdd` daemon runs on the master server and allows the master copy of the password file to be updated remotely using the `yppasswd` command. Digital recommends that you run the `yppasswdd` daemon.

    If you decide to run the `yppasswdd` daemon, edit the `/etc/rc.config` file by issuing the following command:

    ```
    # /usr/sbin/rcmgr set NIS_PASSWDD YES
    ```

    To complete setting up the master server, go to Section 15.3.

15.2 Setting Up Slave Servers

Use the following procedure to set up a slave server:

1. Create the domain directory by entering the following command, replacing `test_domain` with the name that you have chosen for your domain:

   ```
   # mkdir /var/yp/test_domain
   ```

2. Copy the master's maps to the slave server.

   You must copy each map from the master individually, using the following command format:

   /var/yp/ypxfr -h *nis_master* **-c -d** *nis_domain mapname*

   For example, to transfer the `passwd` maps from the master server, `host1`, to a slave server, type the following. The domain in this example is test_domain.

```
# /var/yp/ypxfr –h host1 –c –d test_domain passwd.byname
# /var/yp/ypxfr –h host1 –c –d test_domain passwd.byuid
```

The /var/yp/*nis_domain* directory on the master server lists all of the maps that your slave server can serve.

3. Edit the /var/spool/cron/crontabs/root file with the following entries. Note that there should be no blank lines in the /var/spool/cron/crontabs/root file.

```
   •
   •
   •
# 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_1perday
```

The first line is a comment. The second line runs the following command once an hour at 30 minutes past the hour:

```
sh /var/yp/ypxfr_1perhour
```

The third line runs the following command twice per day at 01:31 and 13:31:

```
sh /var/yp/ypxfr_2perday
```

The fourth line runs the following command once per day at 01:32:

```
sh /var/yp/ypxfr_1perday
```

See the crontab(1) reference page for more information. To complete setting up a slave server, go to Section 15.3.

15.3 Setting Up NIS Clients

This section includes all of the steps necessary to set up an NIS client. Because the master server and all slave servers are considered NIS clients, you must also complete these steps to set up these servers:

1. Prepare the local /etc files.

If you want your system to query an NIS server for password or group information, or both, a plus sign followed by a colon (**+:**) must be the last line of the /etc/passwd file, the /etc/group file, or both. For example:

```
root:9Pf.mMEPU:0:1:System PRIVILEGED Account,,,:/:/bin/csh
field:OnGgTH5mo:0:1:Field Svc Account,,,:/usr/field:/bin/csh
operator:Ni6WK/uqs:25:28:Operator Account,,,:/etc/operator:
guest:Nologin:100:31:Guest account:/usr/spool/uucppublic:/bin/date
+:
```

Note

> If +: is not the last line of the file, all entries following the +: are ignored.

2. Edit the /etc/rc.config file by using the /usr/sbin/rcmgr utility. The syntax for the /usr/sbin/rcmgr command is:

   /usr/sbin/rcmgr set *variable value*

   Digital recommends that you set the value of the NIS_CONF variable and the NIS_ARGS in the /etc/rc.config file to the following values for the master server, slave servers, and clients:

   - NIS_CONF YES
   - NIS_ARGS -S *nisdomain, server1,server2,server3*

   You must set the variable to the system type: MASTER for master servers, SLAVE for slave servers, and CLIENT for clients. The servers must list themselves in the server list, if the system is running with the –S option.

   For example, if you are setting up host2 to be a client server in the domain test_domain, and you want to run the ypbind daemon with the –S option, you could enter the following commands:

   ```
   # /usr/sbin/rcmgr set NIS_CONF YES
   # /usr/sbin/rcmgr set NIS_TYPE CLIENT
   # /usr/sbin/rcmgr set NIS_DOMAIN test_domain
   # /usr/sbin/rcmgr set NIS_ARGS "-S \
      test_domain,host2,host1,host3"
   ```

3. Start the NIS daemons by issuing the following command:

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

If you are reconfiguring NIS on your system, you must first kill the daemons that are running before restarting them. To kill the daemons, enter the following command:

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

Restart the daemons by using the /sbin/init.d/nis start command.

15.4 Post-Setup Procedures

You must edit the /etc/svc.conf file after you configure NIS on your system. The /etc/svc.conf file is the database service selection configuration file that your system references to determine what distributed database lookup services are running on your system, which databases are being served by them, and in what order to query them. After configuring NIS, you must edit the /etc/svc.conf file to tell your system that you want NIS servers queried for distributed database information. For information on editing the /etc/svc.conf file, see Appendix B.

15.5 Adding Users in a Distributed Environment

In an NIS environment you can add a user account to either the local passwd file or the the NIS distributed passwd file. Accounts added to the local passwd file are visible only to the system to which they are added. Accounts added to the NIS distributed passwd file are visible to all NIS clients that have access to the distributed file.

15.5.1 Gathering Information

Before editing the passwd database with new user accounts, gather the following information:

- Determine whether you want to add the account to the local passwd file or the the NIS distributed passwd file.
- Gather the following information on the users you want to add:
  - Login names
  - User identification numbers (UIDs)
  - Group identification numbers (GIDs)
  - Real names, office numbers, and telephone extensions
  - Initial working directories
  - Program to use as a shell

15.5.2 Adding User Accounts to the NIS Distributed passwd File

To add user accounts in a distributed environment, you must edit the master passwd file on the NIS master server. To do this, perform the following steps:

1. Log in as superuser on the NIS master server.
2. Change the directory to /var/yp/src.
3. Edit the passwd database with an entry for each new user.

   The format for each new entry is the same as the format in the /etc/passwd file, which is as follows:

   login-name:passwd field:UID:GID:user-info:initial-working-directory:shell-program

   Leave the *passwd* field blank.
4. Rebuild the passwd database.

   Change the directory to the /var/yp directory and run the make passwd command, as follows:

   ```
   # cd /var/yp
   # make passwd
   updated passwd
   pushed passwd
   ```
5. Use the yppasswd command to set the password for each new user, as follows:

   ```
   # yppasswd new_user
   Old NIS password: Return
   New password: password
   Retype new password: password
   NIS passwd changed on NIS-master
   ```

   Your system is not secure if no password is set.

15.5.3 Adding User Accounts to the Local passwd File in an NIS Distributed Environment

In an NIS environment, if you want to add a user account to the local system only, you must add the account manually. (For more information, see the *System Administration* manual.) Be sure to add these entries prior to the plus sign and colon (+:) at the end of the file.

15.5.4 More Information

For more information on adding users in an NIS environment, see yppasswd(1). For more information on adding user accounts, see the *System Administration* manual.

Manually Setting Up the Network File System 16

This chapter describes how to set up the Network File System (NFS) manually. Setting up NFS includes configuring the following:

- Servers
- Clients, using `/etc/fstab`
- Clients, using `automount`

16.1 Setting Up Servers

Use the following procedure to set up an NFS server:

1. Create the `/etc/exports` file and add the appropriate entries to it.

   The entries that you add are site-specific but their syntax should be as follows:

   pathname [**-root=***0*] [**-root=***hostlist*] [**-anon=***uid*] [**-rw=***hostlist*] [**-ro**]
   [*identifier_1 ... identifier_n*]

   You can use the number sign (#) as a delimiter to add comments. For more information, see `exports`(4) and the *Network Administration and Problem Solving* manual.

2. Add the following information to the `/etc/rc.config` file by using the `/usr/sbin/rcmgr` utility. The syntax for the `/usr/sbin/rcmgr` command is as follows:

   /usr/sbin/rcmgr set *variable value*

   - This system is an NFS server.

     To indicate that this system is a server, enter the following command:

     ```
     # /usr/sbin/rcmgr set NFSSERVING 1
     ```

     A zero (0) in place of the 1 indicates that this system is not a server.

   - The number of `nfsd` daemons that you want the system to run.

     To specify that you want this system to run 12 `nfsd` daemons, enter the following command:

```
# /usr/sbin/rcmgr set NUM_NFSD 12
```

You can run up to 128 nfsd daemons. Although 8 nfsd daemons is usually adequate, if NFS client performance is slow, a possible solution is to increase the number of nfsd daemons.

- Whether to allow users who are not running as root on client systems to mount file systems.

  To indicate that you do not want to allow users on client systems that are not running as root to mount file systems from this server, enter the following command:

  ```
  # /usr/sbin/rcmgr set NONROOTMOUNTS 0
  ```

  To allow users on client systems that are not running as root to mount file systems from this server, enter a 1 instead of 0.

- Whether you want to run the PC-NFS daemon. PC-NFS software provides personal computers on your network with the same capabilities as NFS. PC-NFS is based on the client/server model. The client software runs on the personal computer. The server software runs on the DEC OSF/1 server. Instructions on setting up the PC-NFS client software is provided with the PC-NFS software documentation.

  If you decide to run the PC-NFS daemon, the directory to be mounted on the client should be exported. Also, if network printing is enabled, you should export the /usr/spool/pcnfs directory to the client. For information on exporting directories, see the *Network Administration and Problem Solving* manual.

  To specify that you want to run the PC-NFS daemon, enter the following command:

  ```
  #/usr/sbin/rcmgr set PCNFSD 1
  ```

  You must then export the directories you want to mount on the PC client to the client. Also, you must export the /usr/spool/pcnfs direcotry to the PC client for the client to be able to utilize network printing. For information on exporting directories, see the *Network Administration and Problem Solving* manual.

- Whether you want to run the NFS locking service to allow clients to set advisory record locks on files exported to them. To specify that you want to run the NFS locking service, enter the following

command:

```
#/usr/sbin/rcmgr set NFSLOCKING 1
```

Note, by default, 7 nfsiod daemons are run on all NFS systems. To turn this client service off, enter the following command:

```
#/usr/sbin/rcmgr set NUM_NFSIOD 0
```

The /usr/sbin/rcmgr command appends the information to the end of the /etc/rc.config file. For more information on the rcmgr utility, see rcmgr(8).

3. Make sure that one of the following is true for client systems to which you are exporting file systems:
   - They have an entry in the /etc/hosts file of the server.
   - Their host information is in the hosts database, if the network is serving host information with NIS or BIND.
   - The server specifies the client's Internet address instead of its host name in its /etc/exports file and the mountd daemon is not configured to run with Internet address checking on.
4. Start the NFS daemons by entering the following command:

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

   If you need to stop the NFS daemons, enter the following command:

   ```
   # /sbin/init.d/nfs stop
   ```

16.2 Using the /etc/fstab File to Set Up Clients

Use the following procedure to set up an NFS client, using the /etc/fstab file:

1. Edit the /etc/fstab file.

   Unless you are using automount, the /etc/fstab file must contain an entry for each file system that you want to mount on your system if you want it mounted automatically. You must specify the file system you are mounting, the server you are mounting it from, the permissions with which it is mounted, and the local mount point for it. The syntax for entries in the /etc/fstab file is as follows:

   fs_spec@server fs_file fs_vfstype fs_mntopts fs_freq fs_passno

   For more information, see fstab(4).

The following is a sample /etc/fstab file:

```
/usr/dist@host1  /usr/dist nfs ro,bg 0 0
/usr/share/man@host2   /usr/share/man nfs ro,bg 0 0
/usr/staff/h0@host3 /nfs/host3/usr/staff/h0 nfs rw,bg 0 0
/usr/staff/h1@host3 /nfs/host3/usr/staff/h1 nfs rw,bg 0 0
```

2. Create local mount points.

   You must create a local mount point for each remote file system that you specified in the /etc/fstab file. The local mount points must correspond exactly to the *fs_file* field in the /etc/fstab file. In the preceding example, the client system uses the /etc/fstab file to mount the remote file system /usr/share/man from host2. The /etc/fstab entry specifies that the local mount point is also called /usr/share/man on the client system. While this is the easiest way to name the local mount point, it can have any name. To create the /usr/share/man mount point, enter the following command:

   ```
   # mkdir /usr/share/man
   ```

3. Make sure that one of the following is true for server systems from which you are importing file systems:

   - They have an entry in the /etc/hosts file of the client.
   - Their host information is in the hosts database, if the network is serving host information with NIS or BIND.

4. Edit the /etc/rc.config file by using the /usr/sbin/rcmgr utility. Add the following information to the /etc/rc.config file:

   - Whether this system is an NFS server (a system can be both a client and a server).
   - The number of nfsiod daemons that you want the system to run.

     To specify that you want this system to run 7 nfsiod daemons, enter the following command:

     ```
     # /usr/sbin/rcmgr set NUM_NFSIOD 7
     ```

     You can run up to 20 nfsiod daemons. Although 7 nfsiod daemons is usually adequate, if NFS read and write performance is slow, one possible solution is to increase the number of nfsiod daemons.

   - Optionally, you can turn on the NFS locking service, if you want to be able to set advisory record locks on NFS-mounted files. To do this, enter the following command:

```
# /usr/sbin/rcmgr set NFSLOCKING 1
```

Note that the NFS locking service must also be running on the server.

The /usr/sbin/rcmgr command appends the information to the end of the /etc/rc.config file. For more information on the rcmgr utility, see rcmgr(8).

5. Start the NFS daemons by entering the following command:

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

Note

If you are using automount on this system, you should complete the steps in Section 6.3 before starting the NFS daemons.

If you need to stop the NFS daemons, enter the following command:

```
# /sbin/init.d/nfs stop
```

16.3 Administering Automount Maps

You can customize automount maps to suit your environment and administer them as follows:

- Distribute them using NIS
- Administer them locally
- Distribute them using NIS and administer them locally

For information on creating automount maps and administering them, see Appendix D.

Manually Setting Up the UNIX-to-UNIX Copy Program 17

This chapter describes how to set up the UNIX-to-UNIX Copy Program (UUCP) manually.

Setting up UUCP manually includes the following tasks:

- Checking for required directories
- Optionally, creating the UUCP manager's account
- Creating UUCP accounts for remote systems
- Configuring remote communications links, which consists of editing the following files:
  - `Devices`
  - `/etc/inittab`
  - `Dialers`
  - `Systems`
  - `Dialcodes`
  - `Permissions`
  - `Poll`
  - `remote.unknown`
  - `/etc/inetd.conf`
- Verifying the configuration files
- Setting up TCP/IP communications

Before you set up UUCP, be certain that all of the appropriate hardware is in place. For information on required hardware, see Section 7.1.

17.1 Checking for Required Directories

Verify that the directories, programs, and support files required to operate the UUCP programs are available on the local system. To perform the verification, log in as superuser and enter the `uucheck` command with the –v flag. The `uucheck` program displays an explanation of how it is checking the file structure. If `uucheck` reports any errors, it could indicate that the original software installation process did not complete successfully.

See the *Network Administration and Problem Solving* manual for more information.

17.2 Creating UUCP Accounts for Remote Systems

For a remote system to log in to the local system, it must have an entry in the local `/etc/passwd` file, or it must know the login ID and password for a designated UUCP account on the local system. Likewise, for the local system to log in to a remote system, it must have an entry in the remote system's `/etc/passwd` file, or it must know the login ID and password of a designated UUCP account on the remote system. You must coordinate assigning system login names and passwords for the local `/etc/passwd` file with the system administrator of the remote system.

By convention, the login ID assigned to remote systems is the remote system's name with an uppercase U added as a prefix. Many systems, however, have a single UUCP account for all remote systems to use.

You must add a user account to the `/etc/passwd` file for remote systems that log in to your system.

Use the following procedure to set up a remote system's account:

1. Invoke `vipw` to edit the `/etc/passwd` file as follows:

   ```
   # vipw
   ```

   The syntax for entries in the `/etc/passwd` file is as follows:

   name:password:UID:GID:class:home_dir:shell

   The following is a sample entry for a remote system:

   ```
   Uhost1::4:2:uucp login for host1:/usr/spool/uucppublic:\
       /usr/lib/uucp/uucico
   ```

   For more information, see the `passwd`(4) reference page.

2. Use the `passwd` command to set a password for the new account. Note that the password you supply does not echo to the screen.

   ```
   # passwd Uhost1
   Changing password for Uhost1.
   New password:
   Retype new password:
   ```

The `Permissions` file is used to further control incoming connections and remote systems' access to the local system. For more information on the `Permissions` file, see Section 17.3.6.

17.3 Configuring Remote Communications Links

There are three ways to set up the communications link needed for remote communications:

- Use a hardwired line with a device such as a workstation. The hardwired connection links a port on the local system to a port on the remote system. A hardwired line is advantageous when users on local systems communicate frequently with remote systems; the link is always available and access time is short. However, a port used for a hardwired communications link is not available for any other purpose.

  A hardwired connection is made over an RS-232 or RS-422 serial port at transmission rates of up to 19,200 bits per second. The recommended length of such direct links is 50 feet or less because signal noise becomes a problem with greater distances. It is possible to obtain longer lengths by using a lower transmission rate, limited distance modems (short-haul modems), or both at both ends of the link.

- Use a telephone line with a modem. In this case, the user on the local system establishes the connection to a remote system through an Automatic Calling Unit (ACU), also referred to as an autodialer or a modem. The modem attached to the remote system answers the telephone, and the communications software then completes the connection.

  The advantage of a modem connection using a phone line is that the local and remote ports are not dedicated to a single system. The disadvantage is that the port of the remote system may be busy handling a connection with another system. A dialup link also requires additional software and hardware, such as the ACU, that is not necessary with a hardwired connection.

- Use a TCP/IP connection over a local area network (LAN).

In order for UUCP to function correctly at your site, configure the remote communication facilities by doing the following:

- Edit the `Devices` file and add a list of the devices used to establish a hardwired communications link, a communications link using TCP/IP, or a communications link using a telephone line and a modem.
- Edit the `Dialers` file and add a list of autodialers (modems) used to contact remote systems via the telephone network.
- Edit the `Systems` file and add a list of the remote systems with which the local system can communicate.
- Optionally, edit the `Dialcodes` file and add a list of alphabetic abbreviations representing the prefixes of telephone numbers used to contact the specified remote systems.

- Edit the `Permissions` file and add the appropriate access permissions specifying the way in which local and remote systems can communicate.
- Edit the `Poll` file and add a schedule for monitoring the networked remote systems.

17.3.1 Editing the Devices File

The `Devices` file contains information about hardwired, telephone, and TCP/IP communications links. Each entry in the `Devices` file includes the following fields:

`Type`
Specifies the type of hardwired or autodialer device.

`Line`
Specifies the device name for the port.

`Line2`
Specifies the device name of an 801 ACU (seldom used).

`Class`
Specifies the transmission speed.

`Dialer-Token Pairs`
Specifies a particular type of autodialer (modem) and the token (a defined string of characters) that is passed to the dialer.

Note that all fields must be filled in. If no information is relevant for a particular field enter a dash (–) as a placeholder.

The syntax for entries in the `Devices` file is as follows:

Type Line Line2 Class Dialer-Token Pairs

17.3.1.1 The Type Field

Valid entries for the *`Type`* field include:

`Direct`
Use this keyword, which must begin with an uppercase D, if your site uses hardwired lines to connect multiple systems.

This keyword is used only by the `cu` command for a line to a system or modem. A separate entry must be entered for each dial-out line to be used by `cu`. This does not imply the need for additional dial-out lines, just additional entries in the `Devices` file.

`ACU`
Use this keyword, which you must type in uppercase letters, if your site

connects multiple systems over the telephone network by using ACUs (autodialers, or modems).

`NETWORK`
: Enter TCP (in uppercase letters), if your site uses TCP/IP. (TCP/IP is the only network currently used with UUCP.)

`system_name`
: Enter the name of a particular remote system hardwired to the local system. The *`system_name`* is the name assigned to each host; for example, `host1`.

17.3.1.2 The Line Field

In the *`Line`* field, you should type the device name for the line or port, or the dialer used in the communications link. Use the appropriate device name, for example, `ttyd1`.

17.3.1.3 The Line2 Field

If you entered `ACU` as the keyword in the *`Type`* field, and the autodialer in the `Dialer-Token Pairs` field is a standard 801 dialer, enter the device name of the 801 ACU in the *`Line2`* field. For example, if the type is `ACU` and the line is `ttyd0`, the *`Line2`* entry might be `ttyd1`. If the device type is not 801, you must use a dash (–) in this field as a placeholder. The *`Line2`* field is used only to support older modems that require 801-type dialers. The modem is plugged into one serial port, and the 801 dialer is plugged into a separate serial port.

17.3.1.4 The Class Field

For a hardwired line, type the transmission rate of the device connecting the two systems in the *`Class`* field. For a telephone connection, enter the speed or baud rate at which the ACU transmits data; for example, `300` or `1200`.

Some devices can be used at any speed, in which case you should enter the word `Any` (note the uppercase A). This entry tells UUCP to match any speed requested in the `Systems` file.

17.3.1.5 The Dialer-Token Pairs Field

For a hardwired connection, enter the word `direct` in the *`Dialer-Token Pairs`* field.

For a telephone connection, enter the type of dialer and the token that is passed to that modem in the *`Dialer-Token Pairs`* field. The token is either a telephone number or a predefined string used to reach the dialer.

To specify the dialer entry, enter one of the following:

`DECmodemV32`
: This is a DECmodemV32 modem.

`DEC-dmcl`
: This is a DEC-dmcl language modem.

`hayes`
: This is a Hayes dialer.

`trailb`
: This is a Telebit Trailblazer modem.

`scholar`
: This is a Scholar modem.

`scholar-plus`
: This is a Scholar-Plus modem.

`801`
: This is a standard 801 autodialer, with a separate 212-type or 103-type modem.

OTHER DIALERS
: These are other dialers that you can specify by including the relevant information in the `Dialers` file. Some of these include: Penril (`penril`), Ventel (`ventel`), Rixon (`rixon`), Vadic (`vadic`), and Micom (`micom`).

NETWORK
: This represents a communications network. TCP/IP is the network currently supported by UUCP. Type `TCP` here if you have also used TCP as the keyword in the *Type* field.

Each dialer included as part of a *Dialer-Token Pairs* field in the `Devices` file must also be included as an entry in the `Dialers` file.

The *Token* following the *Dialer* represents either a complete telephone number (\ `D`) or a string defined in the `Dialers` file (\ `T`). If the token represents a complete telephone number, which is the default, leave this part of the *Dialer-Token Pairs* field blank. If it is blank, UUCP uses the telephone number listed in the `Systems` file. Some sites, however, do not include complete telephone numbers in the `Systems` file. Instead, the entry in that file contains only the last four digits of the number, preceded by a dial-code abbreviation. This abbreviation references the beginning of the phone number (for example, a 3-digit exchange number or an access code) contained in the `Dialcodes` file.

Note that it is often more efficient to include the complete telephone number in the `Systems` file.

17.3.1.6 Example Device File Entries

The following example illustrates entries in the `Devices` file for ACU, direct, and TCP links:

```
ACU ttyd0 - 9600 scholar-plus  1
host1 ttyd1 - Any direct  2
TCP - - - TCP  3
Direct ttyd1 - 9600 direct     4
```

1 The *Type* field (`ACU`) indicates that this link is made through a modem. Additional lines with the same type label (in this case, `ACU`) can be included in the file.

The *Line* field indicates that this link is to serial port `ttyd0`.

The *Line2* field contains a dash (–) as a placeholder.

The *Class* field indicates that the transmission rate for this line is 9600 baud.

The *Dialer* part of the *Dialer-Token Pairs* field is specified as a Scholar-Plus modem. The `Token` field is left blank (uses the default `\D` option), which tells UUCP to use the complete number listed in the `Systems` file.

2 The *Type* field (`host1`) indicates that this link is a direct link to the system `host1`.

The *Line* field indicates that this link is to serial port `ttyd1`.

The *Line2* field contains a dash (–) as a placeholder.

The *Class* field (`Any`) indicates that the direct line can transmit at any rate.

The *Dialer-Token Pairs* field (`direct`) indicates that this is a direct connection. No dialer is involved.

3 The *Type* field (`TCP`) indicates that this link is made using the TCP protocol running on a local area network (LAN).

The *Line* field contains a dash (–) as a placeholder. No serial port is associated with a TCP link.

The *Line2* field contains a dash (–) as a placeholder.

The *Class* field contains a dash (–) as a placeholder.

The *Dialer-Token Pairs* field contains `TCP`.

4 The *Type* field (`Direct`) indicates that this link is a direct link.

The *Line* field indicates that this link is to serial port `ttyd1`.

The *Line2* field contains a dash (–) as a placeholder.

The *Class* field indicates that the transmission rate for this line is 9600 baud.

The *Dialer-Token Pairs* field (direct) indicates that this is a direct connection. No dialer is involved.

See the Devices(4) reference page for more information on the Devices file.

17.3.2 Editing the /etc/inittab File

The /etc/inittab file supplies the init program with instructions for creating and running initialization processes. The init command reads the /etc/inittab file each time init is invoked. Each port that you use for incoming UUCP connections should have an entry in the /etc/inittab file.

The format of the /etc/inittab file is as follows:

Identifier:Runlevel:Action:Command #Comments

Identifier
Is a unique ID within this file. It can be up to 14 characters long.

Runlevel
Defines the run levels in which the *Identifier* is processed. *Runlevel* corresponds to a configuration of processes in a system. Each process spawned by the init command is assigned one or more run levels in which it is allowed to exist. UUCP processes exist in run levels 2 and 3.

Action
Informs init how to treat the specified process. The init command recognizes the following entries for the *Action* field:

- respawn
- wait
- once
- boot
- bootwait
- powerfail
- powerwait
- off
- initdefault
- sysinit

For more information on each *Action* entry, see the inittab(4) reference page.

Command

Holds the sh command to be run. It is a 1024-character field.

#Comments

Indicates that this file supports comments. Any text following the number sign is ignored.

A typical entry defining a port shared for both incoming and outgoing calls is as follows:

```
u1:23:respawn:/usr/lib/uucp/uugetty -r -t 30 ttyd1 9600 #UUCP
```

17.3.3 Editing the Dialers File

The Dialers file contains an entry for each autodialer (other than an 801-type dialer or a TCP/IP connection) that can be included in the Devices file. Every modem is listed on a line by itself, and each line includes a series of *expect-send* sequences that specify the initial handshaking that occurs on the communications link before it is ready to send or receive data. In this way, the local and remote systems confirm that they are compatible and configured to transfer data.

The handshaking data are included in a string that tells the cu, ct, or uucico programs the sequence of characters to use to dial out on a particular type of modem.

Each entry in the Dialers file includes the following fields:

Dialer Name

Type of dialer; matches the fifth field, the *Dialer-Token Pairs* field, in the Devices file. If this entry is specified as direct or TCP, leave all other fields in this entry blank. Direct and TCP connections do not require any handshaking or ‘‘dialer negotiations.’’

Dial Tone and Wait Characters

The second field consists of two sets of two characters, for a total of four characters. These characters comprise a translation string. In the phone number of the actual remote modem, the first character in each string is mapped to the second character in that set. This entry generally translates the characters equal sign (=) and dash (–) into whatever the dialer uses for ‘‘wait for dial tone’’ and ‘‘pause.’’ For example, in the second line of the sample file that follows, the equal sign (=) translates into W, and the dash (–) translates into P on a Penril dialer in the phone number.

Handshaking

The handshaking that is usually an *expect-send* sequence of ASCII strings is given in the remainder of the line. This string is generally used to pass telephone numbers to a modem, or to make a connection to another system on the same data switch as the local system. If the match succeeds, the line in the `Dialers` file is interpreted to perform the dialer negotiations.

The following example lists typical entries in the `Dialers` file for commonly used modems:

```
hayes  =,-,   ""  \dAT\r\c OK \pATDT\T\r\c CONNECT
penril =W-P   ""  \d > s\p9\c )-W\p\r\ds\p9\c-) y\c : \E\DP > 9\c OK
ventel =&-%   ""  \r\p \r\p-\r\p-$ <K\D%%\r>\c ONLINE!
vadic  =K-K   ""  \005\p *-\005\p-* D\p BER?  \E\D\e \r\c LINE
scholar  ""    ""  \d\002 Ready \p\T\T! Attached
scholar-plus  ""  ""      \d\002 Ready \pdial\040T\T\r Attached:
DECmodemV32   ""  ""      \d\002 Ready \pdial\040T\T\r Attached:
DEC-dmcl      ""  ""      \d\002 Ready \pdial\040T\T\r Attached:
direct
TCP
```

The following list explains how each entry in the first line of the preceding example affects the action of the dialer:

`hayes`

Specifies the dialer is of type `hayes`. This entry must match the fifth field of one of the entries in the `Devices` file.

`=,–,`

Translates both the equal sign (=) and dash (–) characters in the telephone number to a comma (,), which the `hayes` dialer uses for a pause.

`""`

Indicates to wait for nothing; continue with the rest of the string.

`\dAT\r\c`

Causes a delay, then sends `AT` (the Hayes Attention prefix) followed by a carriage return (`\r`) but not a new line (`\c`).

`OK`

Indicates to wait for OK from the modem, signaling that the first part of the string has executed.

`\pATDT\T\r\c`

Causes a pause for a fraction of a second (`\p`), then sends the dialing command followed by the phone number using dial codes translation (`ATDT\T`), and finishes with a carriage return but not a new line (`\r\c`).

CONNECT

Indicates to wait for CONNECT from the remote modem, signaling that the modems are connected at the baud rate specified in the Devices file.

For more information on the Dialers file, see the Dialers(4) reference page.

If you need to modify this example for use at your site and are unsure about the appropriate entries in the handshaking string, refer to the documentation that accompanied the modems that you are including in the Dialers file.

17.3.4 Editing the Systems File

Each entry in the Systems file represents a remote system with which the local system can communicate. The uucp program cannot establish a communications link between the local computer and a remote computer unless the remote system is configured correctly in this file. Every system that communicates with the local system through uucp must have an entry in the Systems file, regardless of which system initiates the connection.

The entries in the Systems file include:

- The name of the remote system
- The times when users can establish a connection between the local and the remote system
- Whether the connection uses a hardwired, telephone, or TCP/IP communications link
- The speed at which the line transmits data
- The phone number used with a modem
- Information required to log in to the remote system

The syntax for entries in the Systems file is as follows:

System_name Time Caller Class Phone Login

Note that you must have an entry in every field of the Systems file. Use a dash (–) as a placeholder if no other value is appropriate.

17.3.4.1 The System_name Field

The System_name field denotes the name of the remote system. System names should be a maximum of seven characters in length. In order to be compatible with some older systems, such names should include only lowercase characters (or digits).

You can list a specific system in the Systems file more than once. Each additional entry for a specific system represents an alternative communications path that uucp will use in sequential order to try to establish a connection between the local and the remote system.

17.3.4.2 The Time Field

The Time field uses strings that indicate the days of the week and the times of day during which users on the local system can communicate with the specified remote system. For example, the string MoTuTh0800—1730 indicates that local users can contact the specified remote system on Mondays, Tuesdays, and Thursdays between 8:00 a.m. and 5:30 p.m. As indicated in the previous example, day can be a list represented by Mo, Tu, We, Th, Fr, Sa, or Su. Also, you can enter Wk if users can contact the remote system on any weekday, or Any if they can contact the remote system on any day of the week including Saturday and Sunday.

Enter the time at which users can contact the remote system as a range of times, and using 24-hour clock notation (where 0000 is equivalent to midnight). Time ranges can span 0000. For example, if the time range entered is 1800-0600, it means that calls can be made between 6 p.m. and 6 a.m. Conversely, no calls can be made between the hours of 6 a.m. and 6 p.m. Or, if users can communicate with the specified remote system only during morning hours, enter a range such as 0800-1200.

If users can contact the remote computer at any time of day or night, leave the *Time* field blank.

You can include multiple *Time* fields by using a comma (,) as a separator. For example, Wk1800-0600,Sa,Su means that users can contact the remote system on any week day at any time, except between the hours of 6:00 p.m. and 6:00 a.m., and at any time on Saturday and Sunday.

You can enter Never in the *Time* field, indicating that the remote system can never be called. Use Never when the remote system is to initiate all UUCP transactions.

You can also include an optional subfield that specifies the minimum time in minutes between an unsuccessful attempt to reach the remote system and the retry time when UUCP again attempts to communicate with that system. This subfield is separated from the rest of the string by a semicolon (;). For example, Wk1800-0600,Sa,Su;2 indicates that if the first attempt to establish communications fails, UUCP should continue to attempt to contact the remote system at 2-minute intervals. If you include this subfield, it overrides the default retry time.

17.3.4.3 The Caller Field

The keywords for the *Caller* field are ACU for a telephone connection using a modem, *System_name* for a hardwired connection, and TCP for a connection using TCP/IP.

If you use TCP, a subfield associated with the caller field specifies a conversation protocol. The default is g. There are three other conversation protocols, t, e, and f, which you can specify by entering with a comma and the appropriate letter. These protocols are faster and more efficient than the g protocol.

Use either the t or e protocol to communicate with a site running any version of UUCP based on the OSF/1 version.

Use the e and f protocol for a site running a version of UUCP other than OSF/1. Use the t protocol for sites running the Berkeley version of UUCP.

17.3.4.4 The Class Field

The *Class* field indicates the speed at which the specified hardwired or telephone line transmits data. The speed can be 300, 1200, 2400, 9600 baud or higher for a hardwired device or telephone connection.

Unless it is necessary to enter a specific transmission rate in this field, use the keyword Any. This keyword instructs UUCP to match any speed that is appropriate for the ACU or system connection that you specified in the Caller field. For a telephone connection, the rate you enter in this field should correspond to the rate you entered in the Class field in the Devices file for this particular ACU. Enter a dash (–) in this field for a TCP/IP connection.

17.3.4.5 The Phone Field

If you are using a hardwired or TCP connection, enter a dash (–) as a placeholder in the *Phone* field.

If this entry represents a telephone connection using a modem, you can enter the remote modem's phone number in one of the following two ways:

- Enter the complete phone number of the modem. If the system is in your local dialing area, enter the local phone number.

  If the system is not in your local dialing area, include any other numbers required to reach the remote modem; for example, numbers for an outside line, long-distance access codes, area codes, or country codes. This type of entry is the most efficient method of including phone numbers if your site uses a small number of telephone connections. However, if your site includes a large number of remote connections established via a phone line and a modem, enter those numbers in the manner described next.

- Enter an optional alphabetic abbreviation that represents the dialing prefix, and any required access and locality codes, and then enter the phone number. If you choose this method, make certain to also include the dialing prefix in the `Dialcodes` file.

  For example, if your site communicates regularly via modems to several systems that are located at the same remote site, it is convenient to replace complete phone numbers of several remote modems with one dial-code abbreviation. Enter the prefix that represents these numbers, together with the unique part of each modem number for each remote system listed in the `Systems` file.

  Then in the `Dialcodes` file, enter the prefix and the numbers associated with it. Note that you need to enter this prefix in the `Dialcodes` file only once for all the remote modems listed in the `Systems` file.

17.3.4.6 The Login Field

The rest of the line after the *`Phone`* field is the *`Login`* field. The *`Login`* field consists of a string of text called the chat script. The chat script defines the conversation that must take place between the local and remote systems before the remote system can establish a connection. The conversation is defined with a series of *`expect-send`* characters.

The *`expect`* field contains characters that the local system expects to receive from the remote system. Once the local system receives those characters, it sends another string of characters that comprise the *`send`* field. For example, the first *`expect`* field generally contains the remote system's login prompt, and the first *`send`* field generally contains the login ID to be used on the remote system. The second *`expect`* field contains the remote system's password prompt, and the second *`send`* field contains the password to be used on the remote system. The *`expect`* field can include subfields entered in the following form:

expect [*–send–expect*] ...

If the local system does not receive (or cannot read) the first *`expect`* string, it sends its own string (the *`send`* string within the brackets) to the remote system. The local system then expects to receive another *`expect`* string from the remote system. For example, the expect string can contain the following characters:

```
login:--login:
```

The local system expects to receive the string `login:`. If the remote system sends that string and the local system receives it correctly, UUCP goes to the next field in the *`expect—send`* sequence. However, if the local system does not receive `login:`, it sends a null character followed by a new line (signified by the absence of a string between the dash (–) string delimiters), and then expects to receive a second `login:` string from the remote computer.

If the remote system does not send an *expect* string to the local system, you can enter double quotation marks (" "), representing a null string, in the first *expect* field. Also, every time the local system sends a field, it automatically transmits a new line following that *send* field. If you do not want to include this automatic new line, enter `\c` (backslash c) as the last two characters in the *send* string.

There are two special strings you can include in the login sequence. The `EOT` string sends an `EOT` (end-of-transmission) character, and the `BREAK` string attempts to send a `BREAK` character.

You can include the following *expect-send* strings in login fields in the `Systems` file:

`" "`

Expect a null string.

`\N`

Null character.

`\b`

Backspace character.

`\c`

If at the end of a field, suppress the new line that normally follows the characters in a send field. Otherwise, ignore this string.

`\d`

Delay 2 seconds before sending or reading more characters.

`\p`

Pause for approximately 1/4 to 1/2 second.

`\E`

Turn on the echo check (useful in the `Dialers` file).

`\e`

Turn off the echo check (useful in the `Dialers` file).

`\K`

Send a `BREAK` character. This is the same as entering `BREAK`.

`\n`

Newline character.

`\r`

Carriage return.

`\s`
: Space character.

`\t`
: Tab character.

`\\`
: Backslash character.

`EOT`
: `EOT` character. When you enter this string, the system sends two `EOT` newline characters.

`BREAK`
: `BREAK` character.

`\ddd`
: Collapse the octal digits (ddd) into a single character and send that character.

The following example is typical of entries in the `Systems` file:

```
host1  Any  ACU  1200  ch6412  login:--login: uucp  word: \
sysuucp
```

The first field is the name of the remote system (`host1`), the second is the time during which users can reach the remote system (`Any`), the third is the caller (`ACU`) used for the connection, the fourth is the transmission rate (`1200`), the fifth is the phone number of the remote modem (`ch6412`), and the sixth is the login sequence (`login:--login: uucp word: sysuucp`).

17.3.5 Editing the Dialcodes File

The `Dialcodes` file contains dial-code abbreviations and partial phone numbers that complete the telephone entries in the `Systems` file. Defining dial-code abbreviations in the `Dialcodes` file is optional. If your site uses a large number of remote connections established over phone lines and modems, you might want to set up such dial-code abbreviations.

Entries in the `Dialcodes` file contain an alphabetic prefix attached to a partial phone number that might include, for example, access codes, area codes, and exchange numbers. Enter an alphabetic prefix representing the partial phone number, together with the remaining digits of that number, in the `Phone` field in the `Systems` file.

If users at your site communicate regularly using telephone lines and modems to different systems that are located at the same remote site, or to systems located at different remote sites, you might want to use dial-code abbreviations in the `Systems` file. Otherwise, you must enter the complete phone number of each remote modem in that file.

Suppose that it is necessary to dial an access code, an area code, and a phone number in order to reach remote modems at a site with which your users communicate on a regular basis. Rather than typing 15 or more digits for each modem at the remote site in the `Systems` file, you can enter an alphabetic prefix (set up in the `Dialcodes` file) and the remaining digits of the phone number for each remote modem.

The form of the entries in a `Dialcodes` file is as follows:

abv dialing_sequence

The *abv* part of the entry is an alphabetic prefix, containing up to eight letters, that you establish when you set up the dial-code listing. The *dialing_sequence* is composed of all the digits in the number that precedes the phone number.

The following is a sample entry in the `Systems` file:

```
host1 Any ACU 1200  btown4567  login:--login: uucp2  \
       word: leather
```

The following is the relevant entry in the `Dialcodes` file for the dialing prefix `btown`:

```
btown9=1617123
```

You need to enter this prefix only once in the `Dialcodes` file.

To communicate with system `host1`, the user enters the appropriate command and the system name. The modem attached to the local system contacts the modem attached to `host1`, using the number 9=1–617–123–4567 (the dashes are optional). The equal sign (=) is translated by the modem into ‘‘wait for dial tone.’’

For more information, see the `Dialcodes`(4) reference page.

17.3.6 Editing the Permissions File

Each system at your site that uses UUCP requires both a `Systems` and a `Permissions` file entry. The `/usr/lib/uucp/Permissions` file contains information about the ways in which the remote computers listed in the `Systems` file are allowed to carry out `uucico` and `uuxqt` transactions with a local system.

The system manager must set up entries in the `Permissions` file that specify a remote system's login ID, whether that remote system is allowed to send files to and receive files from the local system, and which commands the remote system is permitted to execute on the local system.

17.3.6.1 Permissions File Entries

Each entry in a `Permissions` file is a logical line composed of the basic entry (a login ID or the name of a remote system) plus optional entries separated either by a space or a tab. The basic and optional entries are composed of *`name=value`* pairs; that is, the name of the entry or option followed by an equal sign (=), followed by the value of the entry or option. No spaces are allowed within the pair. Comment lines begin with a number sign (#) and occupy the entire physical line. The backslash character (\) is a continuation character that allows related information that spans several physical lines to be interpreted as a single line. Blank lines are ignored.

Entry types in the `Permissions` file are `LOGNAME`, `MACHINE`, or both.

`LOGNAME`
: Contains the login ID of and access permissions for a remote system that is allowed to conduct `uucico` and `uuxqt` transactions with a local system. `LOGNAME` entries concern operations that occur when a remote system contacts a local system. The calling remote system must be listed in the `Systems` file on the local system.

`MACHINE`
: Contains the names of and access permissions for the remote systems with which the local system is allowed to initiate `uucico` and `uuxqt` transactions. `MACHINE` entries concern operations that occur when a local system contacts a remote system, although the permissions in this entry still apply to the remote system's access to the calling local system.

  A remote system listed in a `MACHINE` entry uses the login ID specified in a `LOGNAME` entry to communicate with a local system.

A `LOGNAME` entry specifies one or more login IDs of remote systems that are permitted to log in to the local system in order to conduct `uucico` and `uuxqt` transactions, and the access permissions for those remote systems. The actual login ID can be any name, although the examples in this chapter use a form of the `uucp` login ID.

Whatever login ID you choose *must* have both a UID and a GID that matches that of the `uucp` UID (usually 4) and GID (usually 2).

The following example shows the simplest and most restrictive `LOGNAME` entry:

```
LOGNAME=uucp
```

This example entry uses the `uucp` login ID, which is generally sufficient for `uucico` and `uuxqt` transactions between local and remote computers at most sites. The entry does not contain any optional *`name=value`* pairs, which means that the remote system's access to the local system is restricted to the following default permissions. However, you can include alternative versions of the uucp login ID if certain computers at your site require different types of permissions when accessing the local system.

- The remote system cannot ask to receive any queued files containing work that users on the local system have requested to be executed on the calling remote system.
- The local system cannot send queued work to the calling remote system when that system has completed its current operations. Instead, the queued work can be sent only when the local system contacts the remote system.
- The remote system cannot send files to (write) or transfer files from (read) any location except the uucp public directory (/usr/spool/uucppublic) on the local system.
- Users on the remote system can run only the default commands on the local system. (The default command set includes only the rmail command, which users implicitly execute by issuing the mail command.)

A name can appear in only one LOGNAME entry. For example, if you have one entry for the uucp login ID, that single entry is sufficient for all remote systems using that login ID. (You list these systems in the MACHINE entry.) You can have additional entries using other forms of the uucp login ID such as uucpa or uucp1 (discussed later in this chapter), but you cannot include another uucp entry.

The following LOGNAME entry includes two login IDs used by remote systems, which are specified in a MACHINE entry, to access the local system host1. Note that both IDs use a form of the uucp login, that they are separated by a colon (:), and that there are no spaces in the entry:

```
LOGNAME=uucp:uucp1
```

The second entry type in a Permissions file is the MACHINE entry. This entry contains the name of the local system, the names of the remote systems with which the local system is allowed to engage in uucico and uuxqt transactions, and the access permissions for those remote systems. The following example shows the simplest kind of MACHINE entry:

```
MACHINE=host1:host2
```

In this example, the local system host1 is permitted to communicate with the remote system host2. Note that the two system names are separated by a colon (:), and that the entry includes no spaces or tab characters. As was the case in the LOGNAME examples, there are no optional *name=value* pairs in this entry, indicating that the remote system's access to the local system is limited to the following actions:

- The remote system can send (write) files only to the local public directory.

- The remote system can execute only those commands in the default command set on the local system, normally just the `rmail` command.
- The remote system cannot ask to receive any local system files queued to run on the calling remote system.
- The local system cannot access (read) any files, except those in the public directory on the local system.

Like a `LOGNAME` entry, a `MACHINE` entry can also include a number of different remote systems. For example:

```
MACHINE=host1:host2:host3:host4
```

17.3.6.2 Permissions File Options

The default access permissions in the `Permissions` file are restrictive. However, the `Permissions` file includes a number of options that enable you to customize your `Permissions` file in such a way that different remote systems are allowed different types of access to the local system when using the UUCP file transport and command execution programs:

`REQUEST`

Permits a remote system to ask to receive any queued files containing work that users on the local system have requested to be executed on that remote system.

The following option permits such requests:

```
REQUEST=yes
```

Including this option makes it easy for remote system users to transfer files to and execute commands on a local system. If security is a consideration, you might want to restrict this access so that the local system retains control of file transfers and command executions initiated by remote systems.

You can include this option in both the `LOGNAME` and `MACHINE` entries in the `Permissions` file.

The default, `REQUEST=no`, indicates that the remote system cannot ask to receive any work queued for it on the local system. In this case, the local system must contact the remote system before files and execute commands queued on the local system can be transmitted to the remote system.

`SENDFILES`

The `SENDFILES` option permits the local system to send queued work to the calling remote system after that remote system has completed its current `uucico` or `uuxqt` operations.

The following option permits 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:

```
SENDFILES=yes
```

You can include this option in a LOGNAME entry in the Permissions file.

The default, SENDFILES=call, specifies that local files queued to run on the remote system are sent only when the local system contacts the remote computer. As was the case with the REQUEST option, security considerations at your site might require that you limit a remote system's access to a local system by using the default value for this option.

READ and WRITE

These options specify the pathnames of locations accessible to the uucico daemon when transferring files to or from the local system. The default location for both the READ and WRITE options is the uucp public directory on the local system:

```
READ=/usr/spool/uucppublic WRITE=/usr/spool/uucppublic
```

You can use the slash (/) that represents the root directory on the local system as the value part of the *name=value* pair in a READ or WRITE option. For example, the following entry specifies that uucico (the daemon that transfers uucp and uux requests) can read from or write to any files on the local system under the root directory that allows access by a user with world access to them:

```
READ=/ WRITE=/
```

The source or destination file or directory must be world readable, world writable, or both. You set these permissions with the chmod command. A user who is not logged in as superuser can take away permissions granted by the READ and WRITE options, but that user cannot grant permissions that are denied by these options.

You can specify more than one path for uucico activities, as in the following entry:

```
WRITE=/usr/spool/uucppublic:/usr/news
```

This entry permits uucico to send files to both the UUCP public directory and the /usr/news directory.

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 decide to specify pathnames in these options, you must enter the pathname for every source and destination. If you enter any pathname in either option, you must also explicitly specify the public directory if you want uucico to be allowed to place files in that location.

You can include READ and WRITE options in both LOGNAME and MACHINE entries.

NOREAD and NOWRITE

These options specify exceptions to the READ and WRITE options.

For example, the following entry permits the remote system to read any file on the local system, except those in the /etc directory and its subdirectories:

```
READ=/ NOREAD=/etc WRITE=/usr/spool/uucppublic
```

The WRITE option in this example allows the remote system to transfer files only to the uucp public directory on the local system.

The NOWRITE option functions in exactly the same way as the NOREAD option; that is, it explicitly specifies directories and files on the local system to which the remote system cannot transfer work.

The specifications you enter with the READ, WRITE, NOREAD, and NOWRITE options can help determine the security of your local system in terms of uucico transactions.

You can include NOREAD and NOWRITE options in both LOGNAME and MACHINE entries.

COMMANDS

The COMMANDS option specifies the commands that remote systems listed in that MACHINE entry can run on the local system.

Caution

The COMMANDS option can jeopardize the security of your system. Use it with extreme care.

The default for this option severely limits the commands that remote systems can run on the local system:

```
COMMANDS=rmail
```

This means that remote systems can run only the rmail command on the local system.

When you enter the COMMANDS option in the MACHINE part of an entry in the Permissions file, the commands you specify in that option override the default. For example, the following entry specifies that the remote systems host2, host3, and host4 can run the rmail

(mail) and print commands on host1, the local system:

```
MACHINE=host1:host2:host3:host4 COMMANDS=rmail:print
```

These commands now comprise the default command set for the remote systems listed in the MACHINE entry.

You can also specify pathnames to those locations on the local system where commands issued by users on remote systems are stored. For example, the following entry indicates that in addition to the rmail command, remote systems can also execute the print command, which is stored in the /bin directory:

```
COMMANDS=rmail:/bin/print
```

This option is useful when the default path of the uuxqt daemon does not include a particular directory where a permitted command resides. The default path of the uuxqt daemon includes only the /bin and /usr/bin directories.

If you want to allow a certain remote system to execute all the available DEC OSF/1 commands on the local system, enter the COMMANDS option with the value ALL:

```
COMMANDS=ALL
```

This specifies that the default command set available to the designated remote system (or for a particular login ID used by a remote system) includes all the available DEC OSF/1 commands.

VALIDATE

To a certain degree, the VALIDATE option verifies the identity of the calling remote computer. Thus, it provides some security when you find it necessary to include commands in the default command set that could potentially cause damage when executed by a remote system on a local system.

Including this option in a LOGNAME entry means that the calling remote system must have a unique login ID and password for file transfers and command executions.

The VALIDATE option is meaningful only when the login ID and password are protected.

For example, the following entries specify that if remote system host2, host3, or host4 attempts to log in to the local system, it must use the login ID uucp and the password associated with that login:

```
LOGNAME=uucp VALIDATE=host2:host3:host4
MACHINE=host1:host2:host3:host4 COMMANDS=ALL
```

Once the remote system is logged in, users on that remote system can run all DEC OSF/1 commands on the local system.

The VALIDATE option links a MACHINE entry, which includes a specified COMMANDS option, to a LOGNAME entry associated with a privileged login. The uucp program requires this validating link because the uuxqt daemon, which executes commands on the local system that have been requested by users on a remote system, is not running while the remote system is logged in and therefore does not know which remote system sent the execution request.

CALLBACK

The CALLBACK option specifies that no uucico transactions will occur until the local system contacts the remote system that is attempting to establish a connection. The following option specifies that the local system must contact the remote system before that remote system can transfer any files to the local system:

```
CALLBACK=yes
```

The default value, CALLBACK=no, is usually sufficient for most sites.

If two systems include the CALLBACK=yes option in their respective Permissions files, they will never be able to communicate with each other.

The CALLBACK option can only be used in LOGNAME entries.

OTHER

The OTHER option represents a system name in a MACHINE entry. It enables you to set up access permissions for remote systems not explicitly specified in the existing MACHINE entries in a Permissions file.

Rather than creating separate MACHINE entries for each of these numerous remote systems, you can set up one entry, with OTHER listed as the MACHINE, that includes the appropriate DEC OSF/1 commands specified in a COMMANDS option entry. Then, when it becomes necessary to change the default command set, you change the list of commands in only one entry rather than in numerous entries. You might also want to specify different (generally more restrictive) option values for these remote systems.

The following is an example of this type of entry:

```
LOGNAME=uucp1
MACHINE=OTHER COMMANDS=rmail:/bin/print:/usr/bin/nroff
```

This entry specifies that all remote systems using the uucp1 login ID that are not included in existing MACHINE entries can run the rmail (mail), print, and nroff commands on the local system.

This example has very restricted access permissions. With the exception of the limited command set, both the LOGNAME and MACHINE entries use the default options that restrict remote systems' uucico and uuxqt transactions with a local system. It is a good idea to restrict access permissions when using the OTHER option, although you can include any of the available MACHINE options.

17.3.6.3 Relating LOGNAME and MACHINE Entries

The following example entry shows the relationship between the LOGNAME and MACHINE entries in a Permissions file:

```
LOGNAME=uucp VALIDATE=host2 REQUEST=yes SENDFILE=yes READ=/ WRITE=/
MACHINE=host1:host2 REQUEST=yes COMMANDS=ALL READ=/ WRITE=/
```

The remote computer host2 can engage in the following uucico and uuxqt transactions with the local system host1:

- The remote system can request that files be sent from the local system.
- The local system can send files to the remote system.
- The remote system can execute all available DEC OSF/1 commands on the local system.
- The remote system can read from and write to all directories and files under the root directory.

In this example entry, files owned by the uucp login ID, such as the Systems file, are accessible by editing programs; for example, ed or vi. This means that a user on host2 can examine and modify the Systems file on host1 if the DEC OSF/1 permission codes specify that the file is writable.

This example entry obviously allows unrestricted access to the local system by the remote system listed in the MACHINE entry. If security is a concern at your site, you should probably set up this type of unrestricted LOGNAME/MACHINE entry on a local system only for a remote computer used by a system administrator or members of the uucp group.

Set up another LOGNAME/MACHINE entry in the local Permissions file for remote machines used by individuals who do not require unlimited access to that local system. Use another version of the uucp login ID in the LOGNAME entry. Then, list the remote systems with restricted access in the MACHINE entry, and include only those commands that general users should execute on the local system. You can combine a LOGNAME and a MACHINE entry into one single entry when both parts include the same options.

For example, consider the following entries:

```
LOGNAME=uucp REQUEST=yes SENDFILE=yes READ=/ WRITE=/
MACHINE=host1:host2 REQUEST=yes COMMANDS=ALL READ=/ WRITE=/
```

Both the LOGNAME and MACHINE entries include the same values for the REQUEST, READ, and WRITE options. You can therefore merge the two parts, as shown in the following example:

```
LOGNAME=uucp MACHINE=host1:host2 REQUEST=yes SENDFILE=yes \
COMMANDS=ALL READ=/ WRITE=/
```

If the line representing an entry is too long to fit on the screen, make the last character in that line a backslash (\ ), which indicates continuation, and then enter the remainder of the entry on the next line.

17.3.7 Editing the Poll File

The `Poll` file contains information specifying when UUCP should poll designated remote computers. This file is used with the `/usr/spool/cron/crontabs/uucp` file, the `uudemon.hour` script, and the `uudemon.poll` script. Together, these files are responsible for initiating automatic calls to remote systems to perform certain maintenance tasks.

Each entry in the `Poll` file contains the name of the remote computer followed by a <TAB> character and a sequence of hourly intervals. The hourly intervals are expressed in digits; the digits should be separated from each other by a space. You must specify interval times as digits between 0 and 23. The following example shows a standard entry in the `Poll` file:

```
host1 <TAB> 0   4   8   12   16   20
```

The digits indicate the hourly intervals at which the local system polls a remote system. The preceding entry instructs the local system to poll the remote system `host1` every 4 hours. Modify the times specified in the `Poll` file depending on the needs at your site.

17.3.8 Editing the remote.unknown File

The `uucp` program executes the `/usr/lib/uucp/remote.unknown` shell script when a remote computer that is not listed in the local `Systems` file attempts to communicate with that local system. The `uucp` program does not permit the unknown remote system to connect with the local system.

Instead, the `remote.unknown` script appends an entry to the `/usr/spool/uucp/.Admin/Foreign` file as shown in the following example entry:

```
FOREIGN=/usr/spool/uucp/.Admin/Foreign
echo "date:call from the system $1"<<FOREIGN
```

Modify this file to fit the needs of your site.

17.4 Verifying the Configuration Files

When the UUCP files are customized for your site, issue the uucheck command again to check for possible errors in the Permissions file. However, remember that the uucheck command does not check file or directory modes, nor does it check for duplicate login or MACHINE names.

Issue the uucheck command with the –v flag in the following manner:

```
# uucheck –v
```

This command provides a detailed explanation of the way that UUCP interprets the Permissions file.

If the uucheck command displays an error message, use the pg command to examine the Permissions file and make sure the entries are correct. Then reissue the uucheck command.

Ensure that all the hosts included in the Systems file on the local system are actually on the UUCP network. Use the uuname command for this task. If all the systems are networked correctly, each system name appears on the list displayed on the screen. The hosts on this list are the systems to which users can send mail.

17.5 Setting Up TCP/IP Communications

If your site uses TCP/IP, you must perform some additional tasks in order for TCP/IP to support UUCP communications.

The uucpd daemon handles communications between UUCP and TCP/IP. This daemon enables users on systems linked over a local area network (LAN) to establish uucp connections to other systems.

Use the following procedure to enable UUCP and TCP/IP to communicate:

1. Check to see whether the /etc/services file includes the following line:

   ```
   uucp          540/tcp          uucpd
   ```

   If it does not, add it to the file.

2. To have the uucpd daemon start automatically each time inetd daemon receives one UUCP request, remove the comment symbol (#) from the following line in the /etc/inetd.conf file:

   ```
   # uucp stream tcp nowait uucp /usr/sbin/uucpd  uucpd
   ```

   Restart the inetd daemon.

3. Be sure that the TCP/IP network between the local and remote systems is working. Issue the ping command in the following way to test that the systems can communicate with one another. Replace rhost1 with the

name of the appropriate remote host.

```
# ping rhost1
```

See the ping(8) reference page for more information.

4. Update the Systems, Devices, and Permissions files in the /usr/lib/uucp directory to include the relevant TCP/IP entries, as follows:

   - To update the Systems file do the following:

     a. Select the appropriate TCP/IP conversation protocol to enter in the TCP caller subfield. There are four kinds of protocols: g, t, e, and f.

        - The g protocol, the default, provides error checking and thus is useful over modem connections. However, it creates a large overhead when running UUCP commands.
        - The t protocol presumes an error-free channel and thus it is not reliable for use with modem connections. You can use the t protocol to communicate with a site running both DEC OSF/1 and Berkeley versions of UUCP.
        - Use the e protocol to communicate with sites running both DEC OSF/1 UUCP and other versions of UUCP. The e protocol is not reliable for modem connections.
        - Use the f protocol to communicate with sites running versions of UUCP other than OSF/1. The f protocol is not reliable for modem connectors.

     b. Add the appropriate entries to the Systems file.

        For example, to connect the local system to system host7 using the default g protocol, enter the following line in the Systems file:

        ```
        host7  Any  TCP  -  -  in:--in: uucp1  word: passuucp
        ```

     c. Replace the send and expect characters in the example Login field with the login prompt, login, password prompt, and password that applies to the remote system to which you are connecting.

        The following example shows how to specify that you are using TCP/IP with the t protocol:

        ```
        host7  Any  TCP,t  -  -  in:--in: uucp1  word: passuucp
        ```

- To update the `Devices` file, do the following:
  a. Enter the following line in the `Devices` file:

  ```
  TCP  -  -  -  TCP
  ```

  b. Specify `TCP` in the *`Caller`* field. Enter dashes (–) in the *`Line`*, *`Line2`*, and *`Class`* fields. Enter `TCP` as the *`Dialer`*. This is done to insure that outgoing calls over TCP/IP are enabled.
- To update the `Permissions` file, enter the appropriate `LOGNAME` and `MACHINE` entries. See Section 17.3.6 for information on editing the `Permissions` file.

Note that you must set up an appropriate login ID and password for any remote system that initiates `uucico` and `uuxqt` activities.

Manually Setting Up the Network Time Protocol 18

Setting up the Network Time Protocol (NTP) manually includes selecting your most accurate time source and then configuring the following:

- Local NTP servers
- NTP clients

You can also choose to set your system time with the `rdate` command, which is explained in Section 18.3.

18.1 Setting Up a Local NTP Server

What you must do to configure a local NTP server depends on your time source. If your time source is Internet NTP servers, see Section 18.1.1. If your time source is a local reference clock, see Section 18.1.2.

18.1.1 Time Source — Internet NTP Servers

Use the following procedure to set up your local NTP servers if your time source is Internet NTP servers:

1. Select three Internet primary or secondary servers for each local NTP server.

   Selecting a different set of Internet servers for each local server is recommended. Secondary servers are usually as reliable and accurate as primary servers. See Section 8.2 for information on obtaining a list of Internet servers.

2. Decide which options you want to run.

   You can chose the `–g` option, the `–l` option, or both:

   - The `–g` option to the `xntpd` daemon allows `xntpd` to correct time differences of more than 1000 seconds between your system and that of your system's NTP servers that occur after the `xntpd` daemon is started. Initial time differences are corrected before the `xntpd` daemon is started by the `ntpdate` command which is run at boot time by the `/sbin/init.d/settime` script. If your system is sensitive to security threats, do not use the `–g` option.

- Normally, NTP logs an initialization message, error messages, status messages, and several other informative messages to `syslog`. The `–l` option specifies that NTP will only log the initialization message and error messages to `syslog`.

3. Edit the `/etc/ntp.conf` file.

   You must add a `peer` entry to the `/etc/ntp.conf` file for each Internet server. Each Internet server must either have an entry in the local `/etc/hosts` file or the hosts file distributed by BIND or NIS. The following `/etc/ntp.conf` file is for a local NTP server that is synchronizing its time with the fictitious Internet time servers `host1`, `host2`, and `host3`. The `version 1` after `host3` indicates that `host3` is running the `ntpd` daemon instead of the `xntpd` daemon. (Servers running DEC OSF/1 run the `xntpd` daemon.) The line `driftfile /etc/ntp.drift` indicates the location of the drift file on this system.

```
#
#  XNTPD Configuration File (template)
#
#
# Specify a filename for the driftfile created by xntpd.
# /etc/ntp.drift is the default.
#
driftfile /etc/ntp.drift
#
#
#
#
# Specify several NTP servers as peers (See the xntpd documentation
# for recommendations on selecting peers).
# NOTE: Be sure to specify version 1 for servers running the ntpd
#       daemon.  For example, if server1 runs ntpd and server2 runs
#       xntpd, the two corresponding entries would be:
#
#               peer server1 version 1      # ntpd server
#               peer server2                # xntpd server
#
#
#
# For further information on configuration options, see the xntpd
# documentation.  If you have a local accurate clock (radio clock, etc),
# you will need to specify further configuration options.
#
peer host1
peer host2
peer host3 version 1
```

4. Edit the `/etc/rc.config` file by using the `/usr/sbin/rcmgr` command. The syntax for the `/usr/sbin/rcmgr` command is as follows:

   /usr/sbin/rcmgr set *variable value*

   To edit the `/etc/rc.config` file and add the required information, enter the following series of commands:

```
# /usr/sbin/rcmgr set XNTPD_CONF YES
# /usr/sbin/rcmgr set XNTP_SERV1 host1
# /usr/sbin/rcmgr set XNTP_SERV2 host2
# /usr/sbin/rcmgr set XNTP_SERV3 host3
# /usr/sbin/rcmgr set XNTPD_OPTS "options"
```

Replace host1, host2, and host3 with the names of the Internet primary or secondary servers that you selected in step 1. Replace options with the options you selected in step 2. You must enclose the options in quotation marks (" ").

5. Start the xntpd daemon with the following command:

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

6. Verify that NTP is working by using the ntpq command:

   ```
   # /usr/bin/ntpq -p
   ```

   For information on monitoring the xntpd daemon and using the ntpq command, see the ntpq(8) reference page.

18.1.2 Time Source — Local Reference Clock

Use the following procedure to set up your local NTP servers if your time source is a local reference clock:

1. Choose one of your local NTP servers to be the local reference clock. The other two local NTP servers can be set up as NTP clients that use the local reference clock and each other as peers.

   For example, if host4, host5, and host6 are the local NTP servers and host4 is the local reference clock, then you should set them up as follows:

   - Set up host5 as an NTP client that specifies host4 and host6 as its local NTP servers
   - Set up host6 as an NTP client that specifies host4 and host5 as its local NTP servers

   Complete steps 3 through 6 only if you are setting up the local reference clock.

2. Decide which options you want to run.

   You can choose the –g option, the –l option, or both:

   - The –g option to the xntpd daemon allows xntpd to correct time differences of more than 1000 seconds between your system and that of your system's NTP servers that occur after the xntpd daemon is started. Initial time differences are corrected before the xntpd daemon is started by the ntpdate command which is run at boot time by the /sbin/init.d/settime script. If your system is sensitive to security threats, do not use the –g option.

- Normally, NTP logs an initialization message, error messages, status messages, and several other informative messages to syslog. The –l option specifies that NTP will only log the initialization message and error messages to syslog.

3. Edit the /etc/ntp.conf file and add the following entry:

   ```
   peer 127.127.1.1
   ```

   This entry allows the local reference clock to run at stratum 1. For more information about local reference clocks, see the ntp.conf(4) reference page.

4. Edit the /etc/rc.config file by using the /usr/sbin/rcmgr command. The syntax for the /usr/sbin/rcmgr command is as follows:

   /usr/sbin/rcmgr set *variable value*

   To edit the /etc/rc.config file and add the required information, enter the following series of commands:

   ```
   # /usr/sbin/rcmgr set XNTPD_CONF YES
   # /usr/sbin/rcmgr set XNTP_SERV1 host4
   # /usr/sbin/rcmgr set XNTP_SERV2 host5
   # /usr/sbin/rcmgr set XNTP_SERV3 host6
   # /usr/sbin/rcmgr set XNTPD_OPTS "options"
   ```

   Replace host4, host5, and host6 with the names of the hosts that you selected to be servers in step 1. Replace options with the options you selected in step 2. You must enclose the options in quotation marks (“ ”).

5. Start the xntpd daemon with the following command:

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

6. Verify that NTP is working by using the ntpq command:

   ```
   # /usr/bin/ntpq –p
   ```

   For information on monitoring the xntpd daemon and using the ntpq command, see the ntpq(8) reference page.

18.2 Setting Up NTP Clients

Use the following procedure to set up an NTP client:

1. Decide which options you want to run.

   You can choose the –g option, the –l option, or both:

- The –g option to the xntpd daemon allows xntpd to correct time differences of more than 1000 seconds between your system and that of your system's NTP servers that occur after the xntpd daemon is started. Initial time differences are corrected before the xntpd daemon is started by the ntpdate command which is run at boot time by the /sbin/init.d/settime script. If your system is sensitive to security threats, do not use the –g option.
- Normally, NTP logs an initialization message, error messages, status messages, and several other informative messages to syslog. The –l option specifies that NTP will only log the initialization message and error messages to syslog.

2. For each client, add a peer entry to the /etc/ntp.conf file for each local NTP server. The following /etc/ntp.conf file is for an NTP client that is synchronizing its time with the local NTP servers: host4, host5, and host6. The line driftfile /etc/ntp.drift indicates the location of the drift file on this system.

```
#
#   XNTPD Configuration File (template)
#
#
# Specify a filename for the driftfile created by xntpd.
# /etc/ntp.drift is the default.
#
driftfile /etc/ntp.drift
#
   •
   •
   •
peer host4
peer host5
peer host6
```

   Remember that each local NTP server that you specify must either have an entry in the client's /etc/hosts file or in a BIND or NIS hosts database that is searched by your system.

3. Edit the /etc/rc.config file by using the /usr/sbin/rcmgr command. The syntax for the /usr/sbin/rcmgr command is as follows:

   /usr/sbin/rcmgr set *variable value*

   To edit the /etc/rc.config file and add the required information, enter the following commands:

```
# /usr/sbin/rcmgr set XNTPD_CONF YES
# /usr/sbin/rcmgr set XNTP_SERV1 host4
# /usr/sbin/rcmgr set XNTP_SERV2 host5
# /usr/sbin/rcmgr set XNTP_SERV3 host6
# /usr/sbin/rcmgr set XNTPD_OPTS "options"
```

Replace host4, host5, and host6 with the names of three local NTP servers for your network. Replace options with the options you selected in step 1. You must enclose the options in quotation marks (" ").

4. Enter the following command to start the xntpd daemon:

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

5. Verify that NTP is working by using the ntpq command:

```
# /usr/bin/ntpq -p
```

For information on monitoring the xntpd daemon and using the ntpq command, see the ntpq(8) reference page.

18.3 Setting Network Time with rdate

For your system to use the rdate command to set its time to the average network time when it starts, you must add an entry for rdate to the /etc/rc.config file.

If your network uses the Network Time Protocol (NTP) time service you might still want to put the rdate entry in the /etc/rc.config file; if NTP hosts are unreachable, the system's time will still be set. If NTP hosts are reachable, the ntpdate command, which runs after the rdate command, will set the time to NTP time before starting the xntpd daemon.

You must use the rcmgr command to edit the /etc/rc.config file. Enter the following command to add an entry for the rdate command to the /etc/rc.config file:

```
# /usr/sbin/rcmgr set RDATE_CONF YES
```

Manually Setting Up the Mail System 19

This chapter describes how to set up and start your DEC OSF/1 mail system manually. This involves stopping and starting the `sendmail` utility, making changes to the `/var/adm/sendmail/sendmail.cf` and `/var/adm/sendmail/`*`hostname`*`.m4` files, and running the `newaliases` command. This chapter also provides information about the four mail utilities included in the DEC OSF/1 operating system, and on the `sendmail` utility.

19.1 Setting Up Your Mail System

Setting up your mail delivery system requires that you understand how the `sendmail` utility works and how to modify the `/var/adm/sendmail/sendmail.cf` file and the `m4` files.

19.1.1 The sendmail Utility

The `sendmail` utility is a general-purpose mail router. It enables a user to send mail to other users on the system and to users on other systems. In most cases, the mail utilities rely on `sendmail` to parse mail addresses and to resolve system aliases. Specifically, when a message is sent, the message goes through the following delivery process:

1. The `mail` utility passes the message to the `sendmail` utility.
2. The `sendmail` utility checks its `aliases` database for full expansion of system names.
3. The `sendmail` utility parses the address of the receiver of the mail according to a set of rules. If the message is going to a user on the same system as the sender, `sendmail` passes the message to the `mail` utility for delivery. If the message is going to a user on a remote system, `sendmail` forwards the message to the `sendmail` utility (or the equivalent utility for systems other than DEC OSF/1) on the remote

system by using one of the following protocols, as specified in the address:

- DECnet

  Used to send mail with DECnet (for example, `host::user`).

- uux

  Used to send mail with the UNIX-to-UNIX Copy Program (UUCP) (for example, `user!decosf`).

- SMTP

  Used to send mail with the Transmission Control Protocol/Internet Protocol (TCP/IP) facility (for example, `user@decosf.dec.com`).

4. Once the message arrives on the correct system, the `sendmail` utility (or equivalent utility) passes the message to the `mail` utility for delivery to the receiver's mailbox.

19.1.2 The sendmail Configuration File

The sendmail configuration file, `sendmail.cf`, contains the instructions for how your mail is sent and delivered, and how it is parsed. This file includes several tunable macros that you might need to modify to suit your environment, and one macro that you should be aware of but cannot modify. For more information, see the `sendmail`(8) reference page.

19.1.3 Using m4 Files

An alternate way to fine tune your configuration is to first run the `mailsetup` script (see Chapter 9). The `mailsetup` script generates the `/var/admin/sendmail/sendmail.m4`, `/var/admin/sendmail/`*`hostname`*`.m4`, and `/var/admin/sendmail/Makefile.cf.`*`hostname`* files. You can then edit the `/var/admin/sendmail/`*`hostname`*`.m4` file, modifying the `define` lines. The file contains comment lines (lines that begin with `dnl`), which provide additional information. For example, the following `define` line specifies that RFC976-style addressing is disabled:

```
define (_RFC976, {})dnl
```

To enable RFC976-style addressing, you would modify the line as follows:

```
define (_RFC976, {T})dnl
```

The `T` enables RFC976-style addressing. When you are finished editing the file, you must change your directory to the `/var/adm/sendmail` directory and issue the following command:

```
# make -f Makefile.cf.hostname:
```

This command generates a *hostname*.cf file. To use the new configuration, copy the *hostname*.cf file to sendmail.cf and restart sendmail by using the /sbin/init.d/sendmail restart command.

For more information, see m4(1) and sendmail.m4(8).

19.1.4 User Configurable Mail Locking

Different mailers use different methods to lock mailbox files. DEC OSF/1 enables you to configure the locking style. To do this, use the /usr/sbin/rcmgr set command to set MAILLOCKING in the /etc/rc.config file.

Valid values for MAILLOCKING are as follows:

- 0 or 4 – Specifies lockf.
- 1 – Specifies lockfile.
- 2 – Specifies Multi-channel Memo Distribution Facility (MMDF). This applies to MH only.
- 5 – Specifies that both lockf and lockfile are used.

19.1.5 Restrictions

Spool files are locked while being modified by using the lockf(3) call and by using a lock file (/var/spool/mail/$ *USER* .lock). When spool files are NFS-mounted the NFS lockd(8) daemon should be running on both the client and server machine. Any user-added program that modifies the spool area must use lockf, the lock file method of locking, or both.

ULTRIX Version 4.3 and earlier versions use lock file locking. Queue files (which reside in the /var/spool/mqueue directory) are locked using lockf(3). Sharing mqueue over NFS is supported with NFS locking (lockd(8)) enabled.

19.2 Starting the Mail System

To start the mail system, use the following procedure:

1. Edit the /var/adm/sendmail/sendmail.cf file to change the macro definitions described in Section 19.1.2.
2. Issue the newaliases command to initialize the sendmail aliases

database as follows:

```
# newaliases
```

3. Stop the current sendmail process, by using the following command:

```
# /sbin/init.d/sendmail stop
```

4. Start the sendmail utility as follows:

```
# /sbin/init.d/sendmail start
SMTP Mail Service started
```

Alternatively, steps 2 through 5 can be accomplished by using the restart option to the sendmail startup script as follows:

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

This command does the following:

- Initializes the sendmail aliases database
- Stops the current sendmail process
- Freezes the sendmail.cf configuration file
- Starts the sendmail utility

19.3 Setting Up the Post Office Protocol

The Post Office Protocol (POP) offers users an alternative to the standard mail system. To enable users on your system to use the Post Offic Protocol for mail, you must enable the mh POP server (popd). To make this option available to users, perform the following steps:

1. Create an account called pop, with /var/spool/pop as the home directory and make pop the owner.
2. Change the owner of /usr/lib/mh/spop to pop by entering the following command:

```
# chown pop /usr/lib/mh/spop
```

3. Create a file in the /var/spool/pop directory called POP.
4. Add an entry into the POP file in the following format for every user who uses mh:

```
user::user:::user@<client_address>::::0
```

5. Run the popaka program for every user entered in the POP database. This produces a string.
6. Enter the string produced from running the popaka program in the systemwide alias file.

7. Run newaliases.
8. Run popd in the background and redirect the output to a null file.

If you are runing in a Network Information Service (NIS) environment, you must perform the following steps to enable users on client machines to reply or send mail so that the return address will be correctly sent to the POP server machine:

1. YP aliases for the POP user should point to the POP server machine.
2. Run quick mailsetup or modify the send mail configuration file to specify the POP server machine.
3. Check the svc.conf to make certain the local, yp aliases are there. If they are not there, add them.

For more information, see the following reference pages: POP(5), pop(8), popaka(8), popd(8), and popwrd(8).

Manually Setting Up the SNMP Agent 20

Setting up the Simple Network Management Protocol (SNMP) Agent manually includes the following tasks:

- Editing the /etc/eca/snmp_pe.conf file
- Editing the /etc/eca/internet_mom.conf file
- Restarting the POLYCENTER Common Agent daemons

20.1 Editing the snmp_pe.conf File

The /etc/eca/snmp_pe.conf file is the configuration file for the snmp_pe daemon. You must edit it to add information about the communities and trap communities you want configured on your system, and to indicate whether or not to disable authentication failure traps.

The following default entry in the /etc/eca/snmp_pe.conf file allows any Network Management Station (NMS) to monitor your system:

```
community public  0.0.0.0 readonly
```

To configure specific communities, remove this entry from your file and replace it with your own entry. Community entries in the /etc/eca/snmp_pe.conf file have the following format:

community *community_name NMS_IP_address* **community-type**

Trap community entries have the following format:

trap *trap_community_name NMS_IP_address*

To disable authentication failure traps, you must add the following entry:

```
no_auth_traps
```

The following is a sample /etc/eca/snmp_pe.conf file with the test1, test2, and test3 communities configured:

```
#
# SNMP network management agent configuration database
#
community       test1  128.45.10.100 readonly
community       test1  16.45.7.110   readonly
community       test2  130.160.4.22  readonly
community        test3    0.0.0.0     readwrite
#
trap             test1  128.45.10.100
```

The test1 community can be monitored by the NMS whose IP address is 128.45.10.100 or by the one whose IP address is 16.45.7.110. The test2 community can be monitored by NMS 130.160.4.22 only. The test3 community can be monitored and managed by any NMS within the test3 community. SNMP traps are sent to the NMS at IP address 128.45.10.100 with a community name of test1.

20.2 Editing the internet_mom.conf File

The /etc/eca/internet_mom.conf file is the configuration file for the TCP/IP Management Object Module (MOM). It contains the following information:

- The physical location of the system on which you are configuring the SNMP Agent (the sysLocation parameter)
- The name of the system administrator (the sysContact parameter)
- The default link polling interval which defines the frequency (in seconds) that the TCP/IP MOM checks the state of each attached TCP/IP interface

The default /etc/eca/internet_mom.conf file contains a value of Unknown for the sysLocation and sysContact parameters, and a value of 60 seconds for the link polling interval. Replace Unknown with information about the physical location of your system and the name of the system administrator. To change the default link polling interval, replace 60 with any positive integer.

The following is a sample /etc/eca/internet_mom.conf file with the sysLocation specified as Blding 3, floor 3, the sysContact specified as Helene Stern, and a default link polling interval:

```
#
# This is the configuration file for the TCP/IP MOM.
# A "#" in the first line indicates a comment.
# A line should not be greater than 1023 characters.
# The first line should contain information about the
# location of the system.
# The second line should contain information about the
# contact person for the system.
# The third line should contain the default Link Polling
# Interval value used internally by the Internet MOM (in seconds).

# sysLocation
Blding 3, floor 3

# sysContact
```

```
Helene Stern

# Link Polling Interval
60
```

20.3 Restarting the POLYCENTER Common Agent Daemons

For the changes to take effect, restart the SNMP Agent daemons using the following commands:

```
# /sbin/init.d/common_agent stop
# /sbin/init.d/common_agent start
```

Some of the variables in the IP routing table and the Exterior Gateway Protocol (EGP) group are obtained from the `gated` daemon, if it is running on the system. If the `gated` daemon is not running prior to starting the Common Agent daemons, the default values are used for these variables.

Configuration Worksheet A

This appendix contains the worksheet that you should fill in before performing the tasks described in this manual. Read the ''Gathering Information'' section of each Chapter and, where appropriate, fill in the blanks. You might want to make a copy of the worksheet for each system that you are setting up. Alternatively, you can obtain a copy of this appendix by printing out the following PostScript file:

```
/usr/examples/network_configuration/worksheet.ps
```

Figure A-1: Configuration Worksheet, Part 1

Part 1: Network Setup

Device name: ______________ ______________

Host name: ______________ ______________

Internet address: ______________ ______________

SLIP remote Internet address: ______________ ______________

Subnet mask: ______________ ______________

Token Ring adaptor speed: ______________ ______________

ifconfig flags: ______________ ______________

SLIP slattach flags: ______________ ______________

SLIP slattach terminal line: ______________ ______________

SLIP slattach baud rate: ______________ ______________

Network daemons

rwhod: Yes □ No □

routed: Yes □ No □ or gated: Yes □ No □

Static Routes (/etc/routes)

□ default gateway □ host □ network

Destination name/IP address: ______________________________

□ gateway □ interface

Name/IP address: ______________________________

/etc/hosts entries

Host name: ______________ ______________

Internet address: ______________ ______________

Alias: ______________ ______________

Host name: ______________ ______________

Internet address: ______________ ______________

Alias: ______________ ______________

/etc/hosts.equiv entries

Host name: ________ ________ ________ ________

Username: ________ ________ ________ ________

/etc/network entries

Network name: ______________ ______________

Network address: ______________ ______________

Alias: ______________ ______________

Figure A-2: Configuration Worksheet, Parts 2 and 3

Part 2: LAT Setup

Number of LAT device special files ______________

Number of getty entries to add to /etc/inittab ______________

Start/stop LAT automatically at boot time Yes □ No □

Part 3: BIND Setup

Domain name: ______________________________

Primary Server

Host name: ______________ ______________

Internet address: ______________ ______________

Secondary Server

Primary server name: ______________________________

Internet address: ______________________________

Slave Server

Server name: ______________ ______________

Internet address: ______________ ______________

Server name: ______________ ______________

Internet address: ______________ ______________

Client

Server name: __________ __________ __________

Internet address: __________ __________ __________

Server name: __________ __________ __________

Internet address: __________ __________ __________

Figure A-3: Configuration Worksheet, Parts 4 and 5

Part 4: NIS Setup

Domain name: ________________

Master Server

Setup options: ________________

Slave name: ________________ ________________

Internet address: ________________ ________________

Slave name: ________________ ________________

Internet address: ________________ ________________

Slave Server

Setup options: ________________

Master name: ________________

Internet address: ________________

Server name: ________________ ________________

Internet address: ________________ ________________

Server name: ________________ ________________

Internet address: ________________ ________________

Client

Setup options: ________________

Server name: ________ ________ ________

Part 5: NFS Setup

Server

Number of nfsd daemons: ________________

Allow nonroot mounts: Yes □ No □

Pathname: ________________ ________________

Network group/Node name: ________________ ________________

Pathname: ________________ ________________

Network group/Node name: ________________ ________________

PCNFS daemon: Yes □ No □ NFS locking: Yes □ No □

Client

Number of nfsiod daemons: ________________

Remote server name: ________________ ________________

Directory path: ________________ ________________

Local mount point: ________________ ________________

Read–only mount: Yes □ No □ Yes □ No □

Automount: Yes □ No □ NFS locking: Yes □ No □

Figure A-4: Configuration Worksheet, Parts 6 and 7

Part 6: UUCP Setup

Connections

Modem type: ____________ ____________ ____________

Baud rate: ____________ ____________ ____________

Device name: ____________ ____________ ____________

inittab entry ID: ____________ ____________ ____________

Outgoing System

Remote system name: ________________________________

Calling times: ________________________________

Phone number: ________________________________

Login ID and password: ________________________________

Incoming System

Remote system name: ________________________________

Local system name: ________________________________

Login ID: ________________________________

Alternative login ID: ________________________________

Options: ________________________________

Part 7: NTP Setup

Server

Time source: ________________________________

Server Internet address: ________________ ________________

Server name: ________________ ________________

NTP daemon: ________________ ________________

Server Internet address: ________________ ________________

Server name: ________________ ________________

NTP daemon: ________________ ________________

Client

Local NTP server address: ________________ ________________

Server name: ________________ ________________

NTP daemon: ________________ ________________

Local NTP server address: ________________ ________________

Server name: ________________ ________________

NTP daemon: ________________ ________________

Figure A-5: Configuration Worksheet, Parts 8 and 9

Part 8: Mail System Setup

Unqualified host name: ______________________________

Host name aliases: ______________________________

Domain aliases: ______________________________

Local domain name: ______________________________

Top level domain name: ______________________________

External TCP format: ______________________________

DECnet:

Node name: ______________________________

Phase IV compatible synonym (optional): ______________________________

Phase IV compatible node number (optional): ______________________________

Namespace: ______________________________

Phase IV domain (for encapsulation): ______________________________

Phase V domain (for encapsulation): ______________________________

Relays:

General: Host name ______________ Protocol ______________

uucp: Host name ______________ Protocol ______________

DECnet: Host name ______________ Protocol ______________

UMC: Host name ______________ Protocol ______________

Part 9: SNMP Setup

System administrator: ______________________________

System location: ______________________________

Link Polling Interval (in seconds): ______________________________

Disable authentication failure traps: Yes □ No □

Community name:	Internet Address:	Community Type:

Trap Community name:	Internet Address:

Setting Up the Database Services Selection File B

If you set up the Network Information Service (NIS), Berkeley Internet Name Domain (BIND) service (for the hosts database only), or both, you must modify the /etc/svc.conf file to reflect the order in which you want these services queried. If you set up NIS or BIND on your system but fail to edit the /etc/svc.conf file correctly, the lookup services are not used.

Because both the bindsetup and nissetup scripts invoke the svcsetup script either noninteractively or interactively, you rarely need to invoke the svcsetup script from the command line. However, if you did not run svcsetup from within bindsetup or nissetup, and you are running BIND, NIS, or both, on your system, you must modify the /etc/svc.conf file, either by using the svcsetup script or manually.

B.1 Gathering Information

Gather the following information before editing the /etc/svc.conf file:

- Whether you are running NIS, BIND, or both.
- Which databases you want served by NIS.
- The order you want your system to search for host information. For example, first search the local /etc/hosts file, then BIND, and then NIS.

The default entry for each database in the /etc/svc.conf file is local. If you have configured NIS, BIND, or both, on your system, you must modify each database entry in the /etc/svc.conf file to reflect the order in which you want the services queried for that database.

Note

> Digital recommends that local be the first service that your system queries for all databases, regardless of what services you are running.

You can use NIS to serve any database. You can use BIND to serve the hosts database only.

B.2 Running svcsetup

Perform the following steps to run the `svcsetup` script:

1. Log in as superuser.
2. Enter the following command to invoke the `svcsetup` script:

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

3. Select the `m` option from the Configuration Menu for the `/etc/svc.conf` file.
4. Enter the numbers from the Change Menu that correspond to the databases whose entries you want to modify.

   For example, to change the lookup service selections for the `group`, `hosts`, and `passwd` databases, enter `1 2 5`:

   ```
      Change Menu for the /etc/svc.conf file

      aliases         => 0
      group           => 1
      hosts           => 2
      netgroup        => 3
      networks        => 4
      passwd          => 5
      protocols       => 6
      rpc             => 7
      services        => 8

      ALL of the above  => 9
        NONE of the above => 10

   Enter your choice(s).  For example "0 3 5" [no default] : 1 2 5
   ```

5. Indicate, for each of the databases that you selected in step 3, the order in which you want the services queried.

   Note

   Selections 3, 4, 5, and 6 are valid for the `hosts` database only.

   ```
   *********************************
   *  Name Service Order Selection *
   *********************************

           local               => 1
           local,yp            => 2
           local,bind          => 3
           bind,local          => 4
           local,bind,yp       => 5
           bind,local,yp       => 6

   Enter the name service order for each of the following
   databases:

    "group" database [2]: Return
    "hosts" database [2]: 5
    "passwd" database [2]: Return
   ```

The svcsetup script indicates that it is updating the /etc/svc.conf file and exits.

B.3 Setting Up the svc.conf File Manually

The distributed database lookup services listed in the /etc/svc.conf file are queried in the order in which they are listed. By default, all entries specify local, meaning that when the system is queried for information it searches the databases on the local system only. You should add yp after local to each of the database entries for which you want NIS servers to be queried. For example, the following sample svc.conf file specifies that local be queried first and then NIS, for all databases except the hosts database. For the hosts database, it specifies that local, then BIND, and then NIS be queried.

```
# @(#)svc.conf    1.0     (DEC OSF/1)    3/10/93
#
# Description:  The svc.conf file is the database services
#               selection file.
#
# Syntax:  database=service,service
#
# database    Database for which the services are being specified.
# service     Distributed database lookup service to query
#             for information; services are queried in the order
#             they are specified.
#
aliases=local,yp
group=local,yp
hosts=local,bind,yp
netgroup=local,yp
networks=local,yp
passwd=local,yp
protocols=local,yp
rpc=local,yp
services=local,yp
```

Additional Information on the Local Area Transport C

This appendix provides sample programs for Local Area Transport (LAT) specialized applications.

C.1 Specialized Application Service

This section contains a sample program (latdate.c) that illustrates a specialized LAT service, which provides a user with the current date and time.

C.1.1 Defining and Establishing the Service

To use the latdate service, you must perform the following steps as superuser:

1. After entering and compiling the latdate.c code, copy the latdate executable to /usr/sbin.
2. Define the latdate service. For example:

   ```
   # latcp -A -a showdate -i "LAT/date service" -o
   ```

3. Add the dedicated tty process entries to the /etc/inittab file. For example:

   ```
   lattty09:3:respawn:/usr/sbin/latdate /dev/tty09 showdate
   ```

Note

You need an /etc/inittab entry for every simultaneous latdate service you want to run. The previous example only allows for one user of the latdate service at any one time.

4. Allow the new `inittab` entries to take effect by entering the following command:

   ```
   # /sbin/init q
   ```

Users can now log in to the terminal server, access the `latdate` service, and get the date and time by entering the following command:

```
LOCAL> connect showdate

Local -010- Session 1 to SHOWDATE on node TINMAN established

      Wed Sep 09 14:46:15 EDT 1992

      Local -218- Connection to SHOWDATE terminated
                  Service user disconnect request
```

C.1.2 Program Listing

This section provides a listing of the `latdate.c` program. If the LAT subset is installed, you can find an online copy of the program in `/usr/examples/lat`.

Example C-1: Listing of the latdate.c Program

```
/*
 * l a t d a t e
 *
 * Description: This sample program illustrates the use of multiple
 *              lat services.  When a user at a terminal connected to
 *              a terminal server issues a "CONNECT showdate" command
 *              the date & time will be printed on his terminal.
 *
 * To compile:  cc -o latdate latdate.c
 *
 */

#include <sys/ioctl.h>
#include <sys/file.h>
#include <sys/termios.h>
#include <dec/lat/lat.h>
#include <signal.h>
#include <errno.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <paths.h>

main( int argc, char *argv[])
{
      int latfd;
      struct termios termios;
      struct latioctl_ttyi ttyi;
      char *tty, *np;
```

Example C-1: (continued)

```
if (argc < 3) {
    perror ("usage: latdate tty service");
    exit(1);
}
tty = (char *) malloc(strlen(argv[1]) + sizeof(_PATH_DEV) + 1);

strcpy(tty, argv[1]);
chown(tty, 0, 0);
chmod(tty, 0622);
/*
 * open LAT line
 */
latfd = open(tty, O_RDWR|O_NONBLOCK);
if (latfd < 0) {
      perror(open);
    exit(1);
}
(void) fcntl(latfd,F_SETFL,fcntl(latfd,F_GETFL,0) &
                                 ~(FNONBLOCK|FNDELAY));
(void) fcntl(latfd, F_SETFD, 0);
/*
 * Bind a service to the tty device
 */
bzero(&ttyi, sizeof(struct latioctl_ttyi));
strcpy(ttyi.li_service, argv[2]);
if (ioctl(latfd, LIOCBIND, &ttyi) < 0) {
    perror(ioctl);
    exit(1);
}
/*
 * get DESTINATION field
 */
(void) ioctl(latfd, LIOCTTYI, &ttyi);
(void) dup2(latfd, 0);
(void) dup2(latfd, 1);
(void) dup2(latfd, 2);
if (latfd > 2)
    (void) close(latfd);
/*
 * set tty flags & mode
 */
tcgetattr(0, &termios);
termios.c_cflag = TTYDEF_CFLAG;
termios.c_iflag = TTYDEF_IFLAG;
termios.c_lflag = TTYDEF_LFLAG;
termios.c_oflag = TTYDEF_OFLAG;
termios.c_cc[VSUSP] = _POSIX_VDISABLE;
tcsetattr(0, TCSAFLUSH, &termios);
```

Example C-1: (continued)

```
        for (np = ttyi.li_service; *np; np++) {
              if (isupper(*np))
                    *np = tolower(*np);
        }
        execl("/bin/date","date",NULL);
        perror("/bin/date");
        exit(1);
}
```

C.2 Program to Replace getty for Special Services

This section contains a sample program (`latdlogin.c`) that replaces the `/usr/sbin/getty` program for each `tty` used as a LAT/dlogin gateway in a local network. It enables users at terminals connected to a terminal server to log in to remote DECnet nodes without having to log in to (or even have accounts on) the local system.

C.2.1 Defining and Establishing the Service

To use the LAT/dlogin gateway, you must install the DECnet software on your system and perform the following steps as superuser:

1. After entering and compiling the `latdlogin.c` code, copy the `latdlogin` executable to `/usr/sbin`.
2. Define the `latdlogin` service by using the `latcp` command. For example:

   ```
   # /usr/sbin/latcp -A -a dloginsvc -i "LAT/dlogin Gateway" -o
   ```

3. Add the dedicated `tty` process entries to the `/etc/inittab` file. For example:

   ```
   latty14:3:respawn:/usr/sbin/latdlogin /dev/tty14 dloginsvc
   ```

   Note

   You need an `/etc/inittab` entry for every simultaneous `latdlogin` service you want to run. The previous example only allows for one user of the `latdlogin` service at any one time.

4. Allow the new `inittab` entries to take effect by entering the following

command:

```
# /sbin/init q
```

Users can now log in from the terminal server and access the LAT/dlogin gateway service, by entering a command similar to the following:

```
LOCAL> connect DLOGINSVC node HOSTNAME destination LOGINHOST
```

In this example, DLOGINSVC is the service name of the LAT/Telnet gateway provided by the local node HOSTNAME, and LOGINHOST is the remote DECnet node to which you want to log in.

C.2.2 Program Listing

This section provides a listing of the latdlogin.c program. If the LAT subset is installed, you can find an online copy of the program in /usr/examples/lat.

Example C-2: Listing of the latdlogin.c Program

```
/*
 * l a t d l o g i n
 *
 * Description: This sample program acts as a LAT to DLOGIN gateway.
 *              With it, a user at a terminal connected to a terminal
 *              server can log into remote DECnet nodes without
 *              having to log into (or even have an account on) the
 *              local system.
 *
 * To compile:  cc -o latdlogin latdlogin.c
 *
 * Setup:       This program requires that DECnet be installed on
 *              your system.  It is necessary to dedicate one or
 *              more LAT ttys to the service.
 *
 *             1. As super user, copy latdlogin to /usr/sbin
 *             2. Add LAT/dlogin Gateway service
 *                # latcp -A -a svcdlgn -i "LAT/dlogin Gateway" -o
 *             3. Add dedicated tty process entry into /etc/inittab
 *                lattty14:234:respawn:/usr/sbin/latdlogin /dev/tty14 \
 *                  dloginsvc
 *              4. Make the new entry to take effect
 *                  # /sbin/init q
 *              5. Login from terminal server
 *                  LOCAL> connect DLOGINSVC node HOSTNAME dest LOGINHOST
 *
 */
#include <sys/ioctl.h>
#include <sys/file.h>
#include <sys/termios.h>
#include <dec/lat/lat.h>
#include <signal.h>
#include <errno.h>
#include <stdio.h>
#include <string.h>
```

Example C-2: (continued)

```
#include <unistd.h>
#include <paths.h>

main( int argc, char *argv[])
{
      int latfd;
      struct termios termios;
      struct latioctl_ttyi ttyi;
      char *tty, *np;

      if (argc < 3) {
         perror ("usage: latdloign tty service");
       exit(1);
         }

      if ((tty = (char *)malloc(strlen(argv[1])+sizeof(_PATH_DEV)+1))
            ==NULL) {
         perror ("malloc() failed, no buffer available");
         exit(1);
         }

      strcpy(tty, argv[1]);
      chown(tty, 0, 0);
      chmod(tty, 0622);

      /*
       * open LAT line */

      if ((latfd = open(tty, O_RDWR|O_NONBLOCK)) < 0) {
         perror(open);
         exit(1);
         }

      if ((fcntl(latfd, F_SETFL, fcntl(latfd, F_GETFL, 0) &
           ~(FNONBLOCK|FNDELAY))) == -1) {
         perror("fcntl() failed at command ,F_SETFL");
         exit(1);
         }

      if ((fcntl(latfd, F_SETFD, 0)) == -1) {
            perror("fcntl() failed at command F_SETFD");
         exit(1);
         }

      /*
       * do the LIOCBIND ioctl */

      bzero(&ttyi, sizeof(struct latioctl_ttyi));
      strcpy(ttyi.li_service, argv[2]);
      if (ioctl(latfd, LIOCBIND, &ttyi) < 0) {
         perror("ioctl() failed at command  LIOCBIND");
         exit(1);
         }

      /*
       * get DESTINATION field
       */
      if ((ioctl(latfd, LIOCTTYI, &ttyi)) < 0) {
         perror("ioctl() failed at command LIOCTTYI");
         exit(1);
         }

      (void) dup2(latfd, 0);
      (void) dup2(latfd, 1);
      (void) dup2(latfd, 2);
      if (latfd > 2)
```

Example C-2: (continued)

```
          (void) close(latfd);
       /*
        * set tty flags & mode */
       if((tcgetattr(0, &termios)) == -1) {
          perror("tcgetattr() failed");
        exit(1);
          }
       termios.c_cflag = TTYDEF_CFLAG;
       termios.c_iflag = TTYDEF_IFLAG;
       termios.c_lflag = TTYDEF_LFLAG;
       termios.c_oflag = TTYDEF_OFLAG;
       termios.c_cc[VSUSP] = _POSIX_VDISABLE;
       if((tcsetattr(0, TCSAFLUSH, &termios)) == -1) {
          perror("tcsetattr() failed at command TCSAFLUSH");
        exit(1);
          }
       (void) signal(SIGINT, SIG_DFL);
       (void) signal(SIGHUP, SIG_DFL);

       for (np = ttyi.li_service; *np; np++) {
          if (isupper(*np))
            *np = tolower(*np);
          }
       execl("/usr/bin/dlogin","dlogin",ttyi.li_service,0);
          perror("/usr/bin/dlogin");
          exit(1);
    }
```

C.3 LAT Host-Initiated Connection

This section contains the sample program, `dial.c`, that illustrates the use of a LAT host-initiated connection. It connects `/dev/tty`*WX* to a Digital Scholar modem that is attached to the port (LAT_PORT) on the DECserver 700 `LAT_SERVER`. After a successful open operation, it autodials a phone number to a host computer and emulates a terminal connected to the host computer.

C.3.1 Defining and Establishing Connection

To use the `dial.c` program, you must define `LAT_SERVER` and `LAT_PORT` by using the `latcp` command. For example:

```
# /usr/sbin/latcp -A -p ttyxx -H LAT_SERVER -R LAT_PORT -Q
```

Access to `/dev/tty`*WX* must be `Read/Write` for the user of the `dial.c` program.

After entering and compiling the `dial.c` code, copy the `dial.c` executable program to `/usr/sbin`.

Users can now dial out from your host, as follows:

```
# /usr/sbin/dial 6037534771 /dev/tty21
```

C.3.2 Program Listing

Example C-3 provides a listing of the `dial.c` program. If the LAT subset is installed, you can find an online copy of the program in `/usr/examples/lat`.

Example C-3: Host-Initiated Connection

```
/*
 * d i a l
 *
 * Description: This sample program illustrates the use of a LAT Host
 *              Initiated Connection.  It connects /dev/ttyxx to a DEC
 *              SCHOLAR modem that is attached to the port "LAT_PORT"
 *              on the DECserver 700 "LAT_SERVER".  After a successful
 *              open, it autodials a phone number to a host computer
 *              and emulates a terminal connected to the host computer.
 *
 * Setup:       Before invoking 'dial', LAT_SERVER and LAT_PORT must be
 *              defined by the latcp command:
 *
 *              # /usr/sbin/latcp -A -p ttyxx -H LAT_SERVER -R LAT_PORT -Q
 *
 *              Access to '/dev/ttyxx' must be Read/Write for the user
 *              of 'dial'.
 *
 * To compile:  cc -o dial dial.c
 *
 * Usage:       /usr/sbin/dial phone_number /dev/ttyxx
 *
 * Comments:    In terminal emulation:
 *                ^](CTRL/]) for escape character
 *                ^]? for help
 *                ^]b to send break signal
 */

#include <stdio.h>
#include <errno.h>
#include <ctype.h>
#include <signal.h>
#include <sys/types.h>
#include <sys/time.h>
#include <sys/file.h>
#include <sys/ioctl.h>
#include <sys/termios.h>

/*
 * For DEC SCHOLAR modem (See SCHOLAR 2400 Modem Owner's Manual)
 * byte 1:     1 (CTRL/A) - autodialer
 * byte 2:     P - pulse dialing  T - tone dialing
 * last byte:    ! - start dialing
 */

u_char nl[20]={0x01, 'T',1,2,3,4,5,6,7,'!'};

int fd;
```

Example C-3: (continued)

```
void nodial();
extern errno;
void resettty();

main(argc,argv)
int argc;
char *argv[];
{
    char buf[BUFSIZ];
    int len, flags;
    struct termios tty_termios;
    /*
     * Open reverse LAT device.
     */
    if ( (fd = open(argv[2],O_RDWR)) < 0 ) {
       perror(argv[0]);
       exit(1);
    }
    /* get current line attributes */
    if ((tcgetattr(fd,&tty_termios) == -1)){
           perror("tcgetattr() failed");
           exit(1);
    }
    /* If CLOCAL happened not to be set, then set it. We need to
     * be in "local" mode to talk to the modem".
     */
    if ((tty_termios.c_cflag & CLOCAL) == 0) {
           tty_termios.c_cflag |= CLOCAL;
           if ((tcsetattr (fd, TCSANOW, &tty_termios) == -1)) {
                  perror("tcsetattr() failed at TCSANOW");
                  exit(1);
           }
    }
    /* turn off O_NONBLOCK, we don't need it any more */
    flags = fcntl (fd, F_GETFL);
    if (flags == -1) {
           perror("fcntl() failed at command F_GETFL");
           exit(1);
    }
    if ((fcntl(fd, F_SETFL, flags & ~O_NONBLOCK) == -1)) {
           perror("fcntl() failed at command F_SETFL");
           exit(1);
    }
    len = strlen(argv[1]);       /* get phone number  */
    strcpy(&nl[2], argv[1]);
    nl[len+2] = '!';             /* ! for start dialing */
    write(0, "Dialing ", 8);     /* print 'Dialing phone#, wait...' */
    write(0, argv[1], len);
    write(0, ", wait... ", 10);
    write(fd, nl, len+3);
    signal(SIGALRM, nodial);     /* Give call 60 seconds to go thru */
    alarm(60);
    read(fd, buf, 80);
    signal(SIGALRM, SIG_IGN);
    read(fd, buf, 80);
    read(fd, buf, 80);
```

Example C-3: (continued)

```
    printf("\n\n%s\n", buf);
    if (buf[0] == 'A' || buf[0] == 'a') {
           alarm(0);
           termmain();
    }
}
void nodial()
{
    char buf[BUFSIZ];     /* Read/write buffer */

    printf("\nDial out failed\n");
    exit(1);
}

/*
 * The remainder of this program is a terminal emulator.
 */

struct sgttyb Isgttyb, sgttyb, sgttyb1;
struct tchars Itchars, tchars1;
struct ltchars Iltchars, ltchars;
int fd, outfile, ret, ret1;
int  readfd, writefd, exception;
struct timeval timeout;

termmain()
{
    char buf[BUFSIZ];
    char *bufptr;
    int on = 1;
    struct termios tty_termios;

    if (ioctl(0, TIOCGETP, &Isgttyb) < 0)
          perror("ioctl() failed at command TIOCGETP");
    if (ioctl(0, TIOCGETC, &Itchars) < 0)
          perror("ioctl() failed at command  TIOCGETC");
    if (ioctl(0, TIOCGLTC, &Iltchars) < 0)
          perror("ioctl() failed at command TIOCGLTC");

    /*
     * Set the terminal into CBREAK | NOECHO | -CRMOD mode so
     * that we can handle character buffering and echo ourselves. We will
     * also disable all special character handling except ^S and ^Q.
     */
    sgttyb = Isgttyb;
    sgttyb.sg_flags |= CBREAK;
    sgttyb.sg_flags &= ~(ECHO | CRMOD);
    if (ioctl(0, TIOCSETP, &sgttyb) < 0)
          perror("ioctl() failed at command TIOCSETP");
    tchars1 = Itchars;
    tchars1.t_intrc = tchars1.t_quitc = tchars1.t_eofc
                    = tchars1.t_brkc = -1;
    if (ioctl(0, TIOCSETC, &tchars1) < 0)
          perror("ioctl() failed at command TIOCSETC");
    ltchars.t_suspc = ltchars.t_dsuspc = ltchars.t_rprntc
                    = ltchars.t_flushc = ltchars.t_werasc
                    = ltchars.t_lnextc = -1;
    if (ioctl(0, TIOCSLTC, &ltchars) < 0)
          perror("ioctl() failed at command TIOCSLTC");

    if (ioctl(fd, TIOCGETP, &sgttyb1) < 0)
```

Example C-3: (continued)

```
      perror("ioctl() failed at command TIOCGETP");
sgttybl.sg_flags |= RAW;
sgttybl.sg_flags &= ~ECHO;
if (ioctl(fd, TIOCSETP, &sgttybl) < 0)
      perror("ioctl() failed at command TIOCSETP");
if (ioctl(fd, FIONBIO, &on) < 0)
      perror("ioctl() failed at command FIONBIO");

if ((tcgetattr(fd,&tty_termios) == -1))
      perror("tcgetattr() failed");

if ((tty_termios.c_cflag & CLOCAL) != 0) {
      tty_termios.c_cflag &= ~CLOCAL;
      if ((tcsetattr (fd, TCSANOW, &tty_termios) == -1))
             perror("tcsetattr() failed at TCSANOW");
}

signal(SIGHUP, resettty);
signal(SIGINT, resettty);
signal(SIGQUIT, resettty);
signal(SIGBUS, resettty);
signal(SIGSEGV, resettty);

printf("escape character: ^];   help: ^]?\r\n\n");
for (;;) {
    readfd = exception = (1 << fd) + (1 << 0);
    errno=0;
    if ((select(fd+1, &readfd, 0, &exception, 0)) > 0) {
         if (readfd & (1 << fd)) {
                if ((ret = read(fd,buf,BUFSIZ)) <= 0) {
                      printf("\nEXIT! ");
                      resettty();
                }
                ret1 = write(0,buf,ret);
                ret -= ret1;
                bufptr = buf + ret1;

                while (ret) {
                       writefd = 1 << 0;
                       select(fd+1, 0, &writefd, 0, 0);
                       if (writefd & (1 << 0)) {
                              ret1 = write(0,bufptr,ret);
                              ret -= ret1;
                              bufptr = bufptr + ret1;
                       }
                }
         }
         if (readfd & (1 << 0)) {
             ret = read(0,buf,BUFSIZ);
                if (*buf == 0x1d) {
                        if ( !(*buf = esccommands()))
                                 continue;
                }
                write(fd,buf,ret);
         }
         if (exception & (1 << fd)) {
                printf("exception: \n");
                printf("\n\nEXIT!\n ");
                 resettty();
         }
    }
    else {
```

Example C-3: (continued)

```
            perror("select: EXIT");
            resettty();
        }
    }
}
void resettty()
{
    int off = 0;

    /*
     * Restore the terminal characteristics to their state before the
     * current session was entered.
     */

    if (ioctl(0, TIOCSETP, &Isgttyb) < 0)
          perror("ioctl() failed at command TIOCSETP");
    if (ioctl(0, TIOCSETC, &Itchars) < 0)
          perror("ioctl() failed at command TIOCSETC");
    if (ioctl(0, TIOCSLTC, &Iltchars) < 0)
          perror("ioctl() failed at command TIOCSLTC");
    close(fd);
    printf("\nDEC OSF/1 LAT dial out disconnected\n\n");
    exit(0);
}

/*
 *        e s c c o m m a n d s
 *
 * for input character:
 * ?:         this menu
 * p:          escape to local command mode
 * b:          send a break
 * esc:     send ^]
 * all others:     exit escape mode
 *
 */
esccommands()
{
    char ch;
    int ret;

    puts("\r\n");
    printf("\r\n\t?\tthis menu\r\n");
    printf("\tp\tescape to local command mode (? for help)\r\n");
    printf("\tb\tsend a break\r\n");
    printf("\tescape\tsend ^]\r\n");
    printf("\tothers\texit escape mode\r\n");
    printf("\nSelect one only - ’p’, ’b’, escape, ’?’   ");
    ret = read(0,&ch,1);
    switch(ch)
    {
    case ’p’:
             localcommands();
             break;

    case ’b’:
             if (ioctl(fd, TIOCSBRK, 0) < 0)
                  perror("ioctl() failed at command TIOCSBRK");
             else
                  printf("\r\nSend a break successfully\r\n");
             break;
```

Example C-3: (continued)

```
    case 0x1b:
             printf("\rYou selected  'escape' \r\n");
             return (0x1d);

    case '?':
             printf("\r\n\t?\tthis menu\r\n");
             printf("\tp\tescape to local command mode (? for help)\r\n");
             printf("\tb\tsend a break\r\n");
             printf("\tescape\tsend ^]\r\n");
             printf("\tothers\texit escape mode\r\n");

    }
    return(0);
}

/*
 *        l o c a l c o m m a n d s
 */
extern char **environ;
localcommands()
{
    char command[512];
    int notdone = 1,pid;

    /*
     * Reset the terminal to its original state.
     */
    if (ioctl(0, TIOCSETP, &Isgttyb) < 0)
          perror("ioctl() failed at command TIOCSETP");
    if (ioctl(0, TIOCSETC, &Itchars) < 0)
          perror("ioctl() failed at command TIOCSETC");
    if (ioctl(0, TIOCSLTC, &Iltchars) < 0)
          perror("ioctl() failed at command TIOCSLTC");

    printf("\r\n\n\t\tLocal Command Menu\r\n\n");
    printf("\tsuspend\tsuspends LAT\n");
    printf("\texit\texits\n");
    printf("\t^D\texits\n");
    printf("\tcmd\tinvoke shell to execute command\n");
    printf("\t\tblank line resumes LAT\n\n");

    printf("\r\n");
    while (notdone) {
       printf("\n\nlocal command> ");
       if (gets(command) == NULL) {
            printf("\nEXIT! ");
            resettty();
       }
       switch (command[0])
       {
           case '?':
                   printf("\tsuspend\tsuspends LAT\n");
                   printf("\texit\texits\n");
                   printf("\t^D\texits\n");
                   printf("\tcmd\tinvoke shell to execute command\n");
                   printf("\t\tblank line resumes LAT\n\n");

           case '\0':
                   notdone = 0;
                   break;

           default:
                  /*
```

Example C-3: (continued)

```
                   * Check for special commands that we handle locally.
                   */
                  if (strcmp(command, "suspend") == 0) {
                         kill(getpid(), SIGTSTP);
                         break;
                  }
                if (strcmp(command, "exit") == 0) {
                         printf("\nEXIT! ");
                         resettty();
                  }
                  if ((pid = fork()) < 0) {
                         perror("LAT server - fork failed");
                         break;
                  }
                  if (pid == 0) {
                         if (execle(getenv("SHELL"), getenv("SHELL"), "-c",
                               command, 0, environ) < 0) {
                               perror("LAT server - unable to exec shell");
                               exit(1);
                         }
                  }

                  wait(0);
                  break;
       }
    }
    /*
     * Reset the terminal to its state on entry.
     */
    if (ioctl(0, TIOCSETP, &sgttyb) < 0)
          perror("ioctl() failed at command TIOCSETP");
    if (ioctl(0, TIOCSETC, &tchars1) < 0)
          perror("ioctl() failed at command TIOCSETC");
    if (ioctl(0, TIOCSLTC, &ltchars) < 0)
          perror("ioctl() failed at command TIOCSLTC");
}
```

Writing automount Maps D

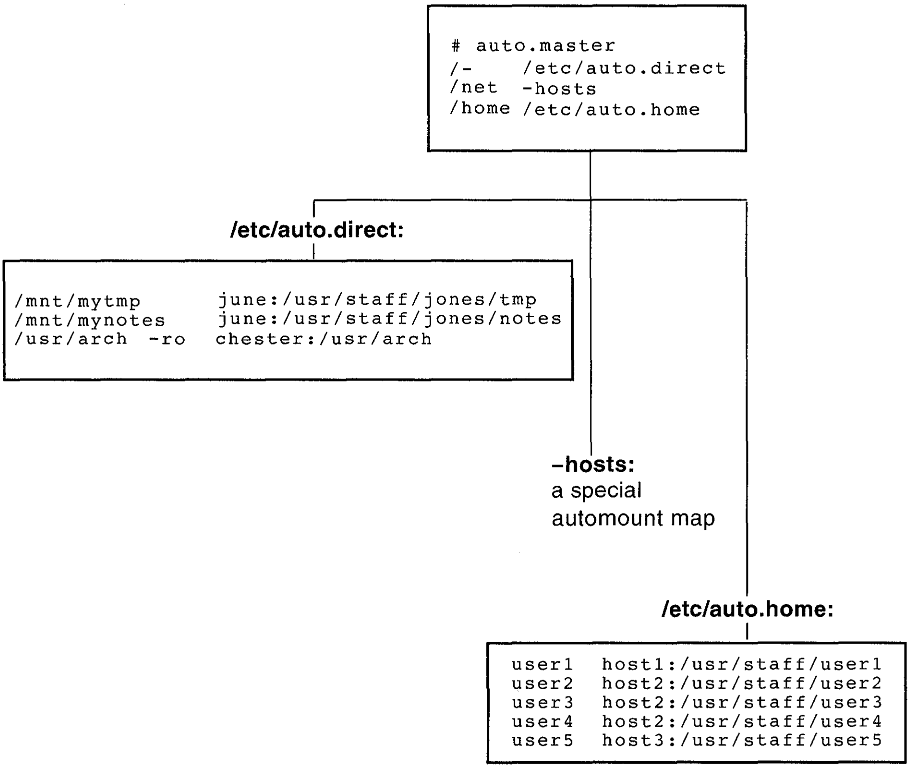
There are three types of automount maps:

- Master
- Direct
- Indirect

The automount maps can be written in a variety of ways. Maps can be direct or indirect. They can be simple or can use multiple mounts, shared mounts, or replicated file systems, or any combination of the three. As discussed in Section D.1, indirect maps can be written to reduce redundancy by using substitution characters and pattern matching. The examples in this section illustrate how the same maps can be rewritten in a number of ways.

Figure D-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. Note that all of the information contained in the master map can be specified on the command line. The master map, however, simplifies organization and administration of automount.

Figure D-1: Sample automount Maps



```
# auto.master
/-     /etc/auto.direct
/net  -hosts
/home /etc/auto.home
```

/etc/auto.direct:

```
/mnt/mytmp        june:/usr/staff/jones/tmp
/mnt/mynotes      june:/usr/staff/jones/notes
/usr/arch  -ro   chester:/usr/arch
```

-hosts:
a special
automount map

/etc/auto.home:

```
user1  host1:/usr/staff/user1
user2  host2:/usr/staff/user2
user3  host2:/usr/staff/user3
user4  host2:/usr/staff/user4
user5  host3:/usr/staff/user5
```

The following examples show how the `/etc/auto.direct` map in Figure D-1 can be rewritten using multiple mounts (Example D-1); multiple mounts and shared mounts (Example D-2); and multiple mounts, shared mounts, and replicated file systems (Example D-3).

Example D-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 D-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 D-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 might read:

```
/mnt      /etc/auto.indirect
```

Rewritten as a simple indirect map (`/etc/auto.indirect`), the `/etc/auto.direct` map in Figure D-1 would read as shown in Example D-4.

Example D-4: Simple Indirect Map

```
mytmp               june:/usr/staff/jones/tmp
mynotes             june:/usr/staff/jones/notes
arch      -ro       chester:/usr/arch
```

Note that the key is a simple pathname.

The following examples illustrate that indirect maps can also be rewritten using multiple mounts (Example D-5); multiple mounts and shared mounts (Example D-6); and multiple mounts, shared mounts, and replicated file systems (Example D-7).

Example D-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 D-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 D-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
```

The –hosts map is a built-in map supplied by automount. This map allows a client to access directories that are exported from any host in its hosts database. The location of the hosts database that your system uses is determined by the services running on your system (BIND, NIS, local) and how those services are specified in the /etc/svc.conf file. References to a particular host name result in all of the file systems that are exported from that host being mounted on the local system. For example, the following command results in all of the file systems that are exported from host1 being mounted on the local system:

```
# cd /net/host1
```

The /etc/auto.home map shown in Figure D-1 is an indirect map that allows users to remote mount their home directories. It can be rewritten using the ampersand (&) and asterisk (*) substitution characters.

The following example shows how the /etc/auto.home map in Figure D-1 can be rewritten using ampersands (&):

```
user1  host1:/usr/staff/&
user2  host2:/usr/staff/&
user3  host2:/usr/staff/&
user4  host2:/usr/staff/&
user5  host3:/usr/staff/&
```

D.1 Substitution and Pattern Matching

The automount daemon recognizes the following substitution characters, allowing you to eliminate redundancy within automount maps:

- Ampersand (&)

  Can be used in both direct and indirect maps; however, it is most efficient and easily understood when used in indirect maps.

- Asterisk (*)

  Can be used in indirect maps only.

Because the ampersand and asterisk are most easily used in indirect maps, this section discusses them in the context of indirect maps only. Recall that lines in indirect maps have the following syntax:

```
key                 mount-options             location
```

Whenever the automount daemon encounters an ampersand (&) in a line of an indirect map, it substitutes the key in that line for the ampersand (&).

The following example is an indirect map that is not using 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. The automount daemon uses the asterisk to match any host not listed as a key in an entry before the asterisk. The following is a typical use of the asterisk (*):

```
#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
```

The automount daemon substitutes the host name (host5) as the key. After it has substituted host5 for the key, it then substitutes host5 for each of the ampersands in the location field as well. The automount daemon translates the preceding command into the following:

```
#key               mount-options            location
#
host5                  -rw,nosuid               host5:/home/host5
```

Note

The automount daemon ignores any entry that follows an asterisk.

D.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
```

The following is an example of a direct map entry that uses the environment variable `HOST` to define subnetworks:

```
/mydir     -rw      server:/export/$HOST
```

D.3 Mounting File Systems

The `automount` daemon provides several ways to mount remote directories and file systems:

- Multiple mounts
- Shared mounts
- Replicated file systems

D.3.1 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 are mounting the `/usr/local` file system on your machine, you can mount the various subdirectories within `/usr/local` from different servers.

The following example could be an entry in a direct map in which the directories `/usr/local/bin`, `/usr/local/src`, and `/usr/local/tools` are mounted from the machines `host1`, `host2`, and `host3`, respectively:

```
/usr/local\
                /bin    -ro       host1:/usr/local/bin \
                /src    -ro       host2:/usr/local/src \
                /tools  -ro       host3:/usr/local/tools
```

This is a direct map because the `key`, `/usr/local`, is an absolute pathname. If this were an entry in an indirect map, the `key` would be a simple pathname, such as `local`. The `key`, `/usr/local`, comprises three subdirectories, each of which is a mount point for a remote directory on a different remote server. The example is displayed showing the entry split into four lines with the continuation lines indented for readability.

The preceding example shows multiple, nonhierarchical mounts under

/usr/local. 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
```

The mount points used here for the hierarchy are /, /bin, /src, and /tools. Note that these mount points are relative to /usr/local. The mount point / mounts /usr/local from host0.

When file systems are mounted hierarchically, the entire hierarchy is treated as one object. Each file system is mounted on a subdirectory within another file system, and when a subdirectory within the hierarchy is referenced, the automount daemon mounts the entire hierarchy. The entire hierarchy is also unmounted as one object.

D.3.2 Shared Mounts

When multiple directories within the same remote directory are mounted, the location field can be specified as follows:

```
host:path:subdir
```

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. This prevents duplicate mounts of the same remote file system when multiple subdirectories within it are accessed. Suppose an indirect map called /auto.myindirect has the following entries:

```
mybin          host1:/usr/staff/diane:bin
mystuff        host1:/usr/staff/diane:stuff
```

When a user accesses a file in /auto.myindirect/mybin, the automount daemon mounts host1:/usr/staff/diane, but creates a symbolic link called /auto.myindirect/mybin to the bin subdirectory in the temporarily mounted file system. If a user immediately tries to access a file in /auto.myindirect/mystuff, the automount daemon needs only to create a symbolic link that points to the mystuff subdirectory because the /usr/staff/diane directory is already mounted. With the following map, the automount daemon would have to mount the file system twice:

```
mybin          host1:/usr/staff/diane/bin
mystuff        host1:/usr/staff/diane/stuff
```

D.3.3 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.

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` command to each of the specified servers. The server that first responds to the `ping` command is used for the mount.

Index

How to Order Additional Documentation

Technical Support

If you need help deciding which documentation best meets your needs, call 800-DIGITAL (800-344-4825) before placing your electronic, telephone, or direct mail order.

Electronic Orders

To place an order at the Electronic Store, dial 800-234-1998 using a 1200- or 2400-bps modem from anywhere in the USA, Canada, or Puerto Rico. If you need assistance using the Electronic Store, call 800-DIGITAL (800-344-4825).

Telephone and Direct Mail Orders

Your Location	Call	Contact
Continental USA, Alaska, or Hawaii	800-DIGITAL	Digital Equipment Corporation P.O. Box CS2008 Nashua, New Hampshire 03061
Puerto Rico	809-754-7575	Local Digital subsidiary
Canada	800-267-6215	Digital Equipment of Canada Attn: DECdirect Operations KAO2/2 P.O. Box 13000 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6
International	————	Local Digital subsidiary or approved distributor
Internal[a]	————	SSB Order Processing – NQO/V19 *or* U. S. Software Supply Business Digital Equipment Corporation 10 Cotton Road Nashua, NH 03063-1260

[a] For internal orders, you must submit an Internal Software Order Form (EN-01740-07).

Reader's Comments

DEC OSF/1
Network Configuration
AA-PS2GB-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

Please rate this manual:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	□	□	□	□
Completeness (enough information)	□	□	□	□
Clarity (easy to understand)	□	□	□	□
Organization (structure of subject matter)	□	□	□	□
Figures (useful)	□	□	□	□
Examples (useful)	□	□	□	□
Index (ability to find topic)	□	□	□	□
Page layout (easy to find information)	□	□	□	□

What would you like to see more/less of? ____________________________________

____

What do you like best about this manual? ____________________________________

____

What do you like least about this manual? ___________________________________

____

Please list errors you have found in this manual:

Page ______ Description ____

______ ____

______ ____

______ ____

______ ____

______ ____

Additional comments or suggestions to improve this manual:

____

____

____

What version of the software described by this manual are you using? ____________

Name/Title ______________________________ Dept. ______________

Company ____ Date ________

Mailing Address ____

____________________ Email ____________________ Phone ______________

---- Do Not Tear – Fold Here and Tape --

digital™

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
OPEN SOFTWARE PUBLICATIONS MANAGER
ZKO3–3/Y32
110 SPIT BROOK ROAD
NASHUA NH 03062–9987

--- Do Not Tear – Fold Here --

Cut
Along
Dotted
Line

Reader's Comments

DEC OSF/1
Network Configuration
AA-PS2GB-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

Please rate this manual:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	□	□	□	□
Completeness (enough information)	□	□	□	□
Clarity (easy to understand)	□	□	□	□
Organization (structure of subject matter)	□	□	□	□
Figures (useful)	□	□	□	□
Examples (useful)	□	□	□	□
Index (ability to find topic)	□	□	□	□
Page layout (easy to find information)	□	□	□	□

What would you like to see more/less of? ______________________

____

What do you like best about this manual? ______________________

____

What do you like least about this manual? ______________________

____

Please list errors you have found in this manual:

Page Description

______ ______________________

______ ______________________

______ ______________________

______ ______________________

______ ______________________

Additional comments or suggestions to improve this manual:

____

____

____

What version of the software described by this manual are you using? __________

Name/Title ______________________ Dept. __________
Company ______________________ Date __________
Mailing Address ______________________
__________ Email __________ Phone __________

---- Do Not Tear – Fold Here and Tape --

digital™

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
OPEN SOFTWARE PUBLICATIONS MANAGER
ZKO3-3/Y32
110 SPIT BROOK ROAD
NASHUA NH 03062-9987

--- Do Not Tear – Fold Here --

**Cut
Along
Dotted
Line**