Mail Administration Guide

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303
U.S.A.

Part No: 805–3480–10
October 1998

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227–14(g)(2)(6/87) and FAR 52.227–19(6/87), or DFAR 252.227–7015(b)(6/95) and DFAR 227.7202–3(a).

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REPONDRE A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.

Contents

Preface

The *Mail Administration Guide* presents the concepts and procedures required to establish and maintain electronic mail services. This book no longer includes information about configuring `sendmail`.

Prior to using this book, you should have already installed Solaris™ software and set up the networking software that you plan to use.

Who Should Use This Book

This book is intended for the system administrator whose responsibilities include setting up and maintaining mail services. Though much of the book is directed toward the experienced system administrator, it also contains information useful to novice administrators and other readers who might be new to the Solaris platform.

How This Book Is Organized

Chapter 1 provides an overview of the mail service. The concepts and terminology necessary to set up and maintain a mail service are discussed in detail.

Chapter 2 describes the steps required to set up and administer a mail service. Troubleshooting tips are included.

Chapter 3 covers special topics related to the Solaris version of `sendmail`.

Related Books

This is a list of related documentation that is refered to in this book.

- *TCP/IP and Data Communications Administration Guide*
- *System Administration Guide, Volume I*
- *System Administration Guide, Volume II*
- Costales, Bryan. *sendmail, Second Edition*. O'Reilly & Associates, Inc., 1997.
- Frey, Donnalyn and Rick Adams. *!%@:: A Directory of Electronic Mail Addressing and Networks*. O'Reilly & Associates, Inc., 1993.

Other Information Sources

Here is a list of additional information sources about `sendmail`.

- `http://www.sendmail.org`— Home page for `sendmail`
- `http://www.sendmail.org/faq`— FAQ for `sendmail`
- `http://www.sendmail.org/m4/readme.html`— README for new `sendmail` configuration files
- `http://www.sendmail.org/sun-specific/migration+sun.html`— Differences between `sendmail` delivered with the 2.6 release and release 7.

Ordering Sun Documents

The SunDocs℠ program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals using this program.

For a list of documents and how to order them, see the catalog section of the SunExpress™ Internet site at `http://www.sun.com/sunexpress`.

What Typographic Changes Mean

Table P–1 describes the typographic changes used in this book.

TABLE P–1 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% You have mail.`

`AaBbCc123`	What you type, contrasted with on-screen computer output	`machine_name%` **`su`**
		`Password:`

AaBbCc123	Command-line placeholder: replace with a real name or value	To delete a file, type `rm` *filename*.
AaBbCc123	Book titles, new words or terms, or words to be emphasized	Read Chapter 6 in *User's Guide*. These are called *class* options. You *must* be root to do this.

Shell Prompts in Command Examples

Table P–2 shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P–2 Shell Prompts

Shell	Prompt
C shell prompt	`machine_name%`
C shell superuser prompt	`machine_name#`
Bourne shell and Korn shell prompt	`$`
Bourne shell and Korn shell superuser prompt	`#`

CHAPTER **1**

Understanding Mail Services

Setting up and maintaining an electronic mail service are complex tasks, critical to the daily operation of your network. As network administrator, you might need to expand an existing mail service or perhaps set up mail service on a new network or subnet. To help you plan a mail service for your network, this chapter provides conceptual information about mail services and briefly outlines the tasks required for setting up typical mail configurations.

Mail Services Terminology

In addition to the mail files and programs, many other components are required to establish a mail service. The following sections define these components and some of the terminology used to describe them.

The first section defines the terminology used when discussing the software parts of the mail delivery system. The next section focuses on the functions of the hardware systems in a mail configuration.

Mail Services Software Terminology

This section describes the software components of a mail system. Each service includes at least one of each of the following:

- Mail user agent
- Mail transfer agent
- Mail delivery agent

Other software components include domain names, mail addresses, mailboxes, and mail aliases.

Mail User Agent

The *mail user agent* is the program that acts as the interface between the user and mail transfer agent, such as the `sendmail` program. The mail user agents supplied with the Solaris operating environment are `/usr/bin/mail`, `/usr/bin/mailx`, `$OPENWINHOME/bin/mailtool`, and `/usr/dt/bin/dtmail`.

Mail Transfer Agent

The *mail transfer agent* is responsible for the routing of mail messages and resolution of mail addresses. This is also known as a mail *transport* agent. The transfer agent for the Solaris operating environment is `sendmail`. The transfer agent performs these functions:

- Accepts messages from the mail user agent
- Resolves destination addresses
- Selects a proper delivery agent to deliver the mail
- Receives incoming mail from other mail transfer agents

Mail Delivery Agent

A *mail delivery agent* is a program that implements a mail delivery protocol. The following mail delivery agents are provided with the Solaris operating environment:

- The UUCP mail delivery agent, which uses `uux` to deliver mail.
- The local mail delivery agent, which is `mail.local` in the standard Solaris release.

Mailers

A *mailer* is a `sendmail` specific term. You can customize a mail delivery agent. A mailer is used by `sendmail` to identify a specific instance of a customized mail delivery agent or a mail transfer agent.

You need to specify at least one mailer in the `sendmail.cf` file of all systems in your network.

The smtp mailer uses SMTP to transfer a message. SMTP is the standard mail protocol used on the Internet. This is an example of an SMTP mail header:

```
To: paul@phoenix.stateu.edu
From: Iggy.Ignatz@eng.acme.com
```

If mail is sent between two users in the same domain, the header looks like this:

```
To: Irving.Who@eng.acme.com
From: Iggy.Ignatz@eng.acme.com
```

Use SMTP for sending mail outside your domain, especially for mailboxes that you must reach through the Internet.

The uucp-old mailer uses uux to deliver messages, but it formats headers with a domain-style address, and the To: and Cc: lines are formatted by domain, much like the SMTP headers. The uucp headers look like this:

```
To: paul@phoenix.stateu.com
From: ignatz@eng.acme.com
```

Use uucp-uudom for UUCP mail to systems that can handle and resolve domain-style names. The sender also must be able to handle domain-style names and be able to receive replies from the Internet.

The uucp-old mailer uses an exclamation point address in the headers. This is one of the original mailers. The headers look like this:

```
To: edu!stateu!phoenix!paul
From: acme!ignatz
```

You can define other mail delivery agents by providing a mailer specification in the sendmail.cf file. Additional information about mailers can be found in /usr/lib/mail/README.

Domain Names

A *domain* is a directory structure for network address naming. Electronic-mail addressing also uses domains. An email address has this format:

```
user@subdomain.  ...  .subdomain2.subdomain1.top-level-domain
```

The part of the address to the left of the @ sign is the local address. The local address can contain information about:

- Routing using another mail transport (for example, `bob::vmsvax@gateway` or `smallberries%mill.uucp@gateway`)
- An alias (for example, `iggy.ignatz`)

The receiving mailer is responsible for determining what the local part of the address means.

The part of the address to the right of the @ sign shows the domain address where the local address is located. A dot separates each part of the domain address. The domain can be an organization, a physical area, or a geographic region.

Domain addresses are case insensitive. It makes no difference whether you use uppercase, lowercase, or mixed-case letters in the domain part of an address.

The order of domain information is hierarchical—the more local the address, the closer it is to the @ sign.

The larger the number of subdomains, the more detailed the information that is provided about the destination. Just as a subdirectory in a file-system hierarchy is considered to be inside the directory above, each subdomain in the mail address is considered to be inside the location to its right.

Table 1–1 shows the top-level domains.

TABLE 1–1 Top-level Domains

Domain	**Description**
`Com`	Commercial sites
`Edu`	Educational sites
`Gov`	United States government installations
`Mil`	United States military installations
`Net`	Networking organizations
`Org`	Other organizations

!%@:: A Directory of Electronic Mail Addressing and Networks by Donnalyn Frey and Rick Adams (O'Reilly & Associates, Inc., 1993) contains a complete list of international top-level domain addresses; it is updated periodically.

For mail delivery, the name space domain name and the mail domain name occasionally do not match. However, the DNS domain name and the mail domain name must be identical. By default, the `sendmail` program strips the first component from the domain name to form the mail domain name. For example, if a NIS+ domain name were `bldg5.eng.acme.com`, its mail domain name would be `eng.acme.com`.

Note - Although mail domain addresses are case insensitive, the name space domain name is not. For best results use lowercase characters when setting up the mail and name space domain names.

Mail Address

The *mail address* contains the name of the recipient and the system to which the mail message is delivered.

When you administer a small mail system that does not use a name service, addressing mail is easy: login names uniquely identify users.

When, however, you are administering a mail system that has more than one system with mailboxes, one or more domains, or when you have a UUCP (or other) mail connection to the outside world, mail addressing becomes more complex. Mail addresses can be *route independent*, *route based*, or a mixture of the two. Route-based addressing is based on old specifications and is not required or desired in most situations.

Route-Independent Addressing

Route-independent addressing requires the sender of an email message to specify the name of the recipient and the final destination address. Route-independent addresses usually indicate the use of a high-speed network like the Internet. In addition, newer UUCP connections frequently use domain-style names. Route-independent addresses can have this format:

```
user@host.domain
```

UUCP connections can use the following address format:

```
host.domain!user
```

The increased popularity of the domain hierarchical naming scheme for computers is making route-independent addresses more common. In fact, the most common route-independent address omits the host name and relies on the domain name service to properly identify the final destination of the email message:

```
user@domain
```

Route-independent addresses are read by searching for the `@` sign, then reading the domain hierarchy from the right (the highest level) to the left (the most specific address to the right of the `@` sign).

Route-Based Addressing

Route-based addressing requires the sender of an email message to specify the local address (typically, a user name) and its final destination, as well as the route that the message must take to reach its final destination. Route-based addresses were fairly common on UUCP networks, and have this format:

```
path!host!user
```

Whenever you see an exclamation point as part of an email address, all (or some) of the route was specified by the sender. Route-based addresses are always read from left to right.

For example, an email address that looks like this:

```
venus!acme!sierra!ignatz
```

means that mail sent to the user named `ignatz` is first sent to the system named `venus`, next to `acme`, and then to `sierra`. (Notice that this is an example and not an actual route.) If any of the mail handlers is out of commission, the message will be delayed or returned as undeliverable.

Mail sent through the `uucp` mailer is not restricted to using route-based addressing. Some `uucp` mailers also handle route-independent addressing.

Mailbox

A *mailbox* is a file on a mail server that is the final destination for email messages. The name of the mailbox can be the user name or a place to put mail for someone with a specific function, like the postmaster. Mailboxes are in the `/var/mail/`*username* file, which can exist either on the user's local system or on a remote mail server. In either case, the mailbox is on the system to which the mail is delivered.

Mail should always be delivered to a local file system so that the user agent can pull mail from the mail spool and store it readily in the local mailbox. Do not use

NFS-mounted file systems as the destination for a user's mailbox. Specifically, do not direct mail to a mail client that is mounting the /var/mail file system from a remote server. Mail for the user, in this case, should be addressed to the mail server and not to the client host name. NFS-mounted file systems can cause problems with mail delivery and handling. Clients that NFS-mount /var/mail go into "remote mode" and should arrange to have the server send and receive mail for them.

The /etc/mail/aliases file and name services like NIS and NIS+ provide mechanisms for creating aliases for electronic mail addresses, so that users do not need to know the precise local name of a user's mailbox.

Table 1–2 shows some common naming conventions for special-purpose mailboxes.

TABLE 1–2 Conventions for the Format of Mailbox Names

Format	Description
username	User names are frequently the same as mailbox names.
Firstname.*Lastname* *Firstname_Lastname* *Firstinitial*.*Lastname* *Firstinitial_Lastname*	User names can be identified as full names with a dot (or an underscore) separating the first and last names, or by a first initial with a dot (or an underscore) separating the initial and the last name.
postmaster	Users can address questions and report problems with the mail system to the postmaster mailbox. Each site and domain should have a postmaster mailbox.
MAILER-DAEMON	sendmail automatically routes any mail addressed to the MAILER-DAEMON to the postmaster.
aliasname-request	Names ending in -request are administrative addresses for distribution lists. This address should redirect mail to the person who maintains the distribution list.
owner-*aliasname*	Names beginning with owner- are administrative addresses for distribution lists. This address should redirect mail to the person who handles mail errors.

TABLE 1–2 Conventions for the Format of Mailbox Names *(continued)*

Format	Description
`owner-owner`	This alias is used when there is no `owner-`*aliasname* alias for errors to be returned to. This address should redirect mail to the person who handles mail errors and should be defined on any system that maintains a large number of aliases.
local%domain	The percent sign (`%`) marks a local address that is expanded when the message arrives at its destination. Most mail systems interpret mailbox names with `%` characters as full mail addresses. The `%` is replaced with an `@`, and the mail is redirected accordingly. Although many people use the `%` convention, it is not a formal standard. It is referred to as the "percent hack." This feature is often used to help debug mail problems.

Starting with version 8, the envelop sender for mail sent to an group alias is changed to the address expanded from the owner alias, if an owner alias exists. This change allows for any mail errors to be sent to the alias owner rather than being returned to the sender. What users will notice is that mail they send to an alias, when delivered, will look like it came from the alias owner. The following alias format will help with some of the problems associated with this change:

```
mygroup: :include:/pathname/mygroup.list
owner-mygroup: mygroup-request
mygroup-request: sandys, ignatz
```

In this example, the `mygroup` alias is the actual mail alias for the group; the `owner-mygroup` alias receives error messages; and the `mygroup-request` alias should be used for administrative requests. This structure means that in mail sent to the `mygroup` alias, the envelope sender changes to `mygroup-request`.

Aliases

An *alias* is an alternate name. For electronic mail, you can use aliases to assign a mailbox location or to define mailing lists.

For large sites, the mail alias typically defines the location of a mailbox. Providing a mail alias is like providing a mail stop as part of the address for an individual at a large corporation. If you do not provide the mail stop, the mail is delivered to a central address. Extra effort is required to determine where within the building the mail is to be delivered, and the possibility of error increases. For example, if two people named Kevin Smith are in the same building, only one of them will get mail.

Use domains and location-independent addresses as much as possible when you create mailing lists. To enhance portability and flexibility of alias files, make your alias entries in mailing lists as generic and system-independent as possible. For example, if you have a user named ignatz on system mars, in domain eng.acme.com, create the alias ignatz@eng instead of ignatz@mars. If user ignatz changes the name of his system but remains within the engineering domain, you do not need to update alias files to reflect the change in system name.

When creating alias entries, type one alias per line. You should have only one entry that contains the user's system name. For example, you could create the following entries for user ignatz:

```
ignatz: iggy.ignatz
iggyi: iggy.ignatz
iggy.ignatz: ignatz@mars
```

You can create an alias for local names or domains. For example, an alias entry for user fred who has a mailbox on the system mars and who is in the domain planets could have this entry in the NIS+ aliases table:

```
fred: fred@planets
```

When creating mail lists that include users outside your domain, create the alias with the user name and the domain name. For example, if you have a user named smallberries on system privet, in domain mgmt.acme.com, create the alias as smallberries@mgmt.acme.com.

The email address of the sender is now automatically translated to a fully qualified domain name when mail goes outside the user's domain.

Uses for Aliases Files

You create mail aliases for global use in the NIS+ mail_aliases table, the NIS aliases map, or in local /etc/mail/aliases files. You can also create and administer mailing lists using the same alias files.

Depending on the configuration of your mail services, you can administer aliases by using the NIS or NIS+ name service to maintain a global aliases database or by updating all the local /etc/mail/aliases files to keep them synchronized.

Users can also create and use aliases. They can create aliases either in their local ~/.mailrc file, which only they can use, or in their local /etc/mail/aliases file, which can be used by anyone. Users cannot normally create or administer NIS or NIS+ alias files.

Hardware Components of a Mail Configuration

A mail configuration requires three elements, which can be combined on the same system or provided by separate systems:

- A mail host
- At least one mail server
- Mail clients

When you want users to communicate with networks outside your domain, you must also add a fourth element, a mail gateway.

Figure 1–1 shows a typical electronic mail configuration, using the three basic mail elements plus a mail gateway. Each element is identified and described in the following sections.

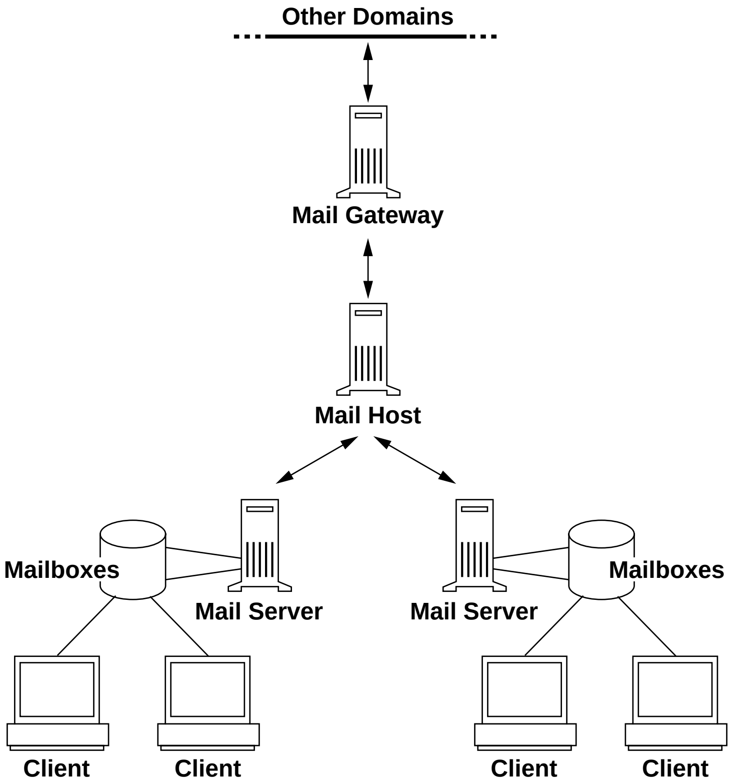

Figure 1–1 Typical Electronic Mail Configuration

Mail Host

A *mail host* is the machine that you designate as the main mail machine on your network. It is the machine to which other systems at the site forward mail that they cannot deliver. You designate a system as a mail host in the `hosts` database by adding the word `mailhost` to the right of the IP address in the local `/etc/hosts` file or in the hosts file in the name service. You must also use the `main.cf` file as the mail-configuration file on the mail host system.

A good candidate for mail host is a system on the local-area network that also has a modem for setting up PPP or UUCP links over telephone lines. Another good candidate is a system configured as a router from your network to the Internet global network. (See *TCP/IP and Data Communications Administration Guide* for more information on PPP, UUCP, and routers.) If none of the systems on your local network has a modem, designate one as the mail host.

Some sites use standalone machines that are not networked in a time-sharing configuration; that is, the standalone machine serves terminals attached to its serial ports. You can set up electronic mail for this configuration by treating the standalone system as the mail host of a one-system network.

Mail Server

A *mailbox* is a single file that contains email for a particular user. Mail is delivered to the system where the user's mailbox resides: the local machine or a remote server. A *mail server* is any system that maintains user mailboxes in its `/var/mail` directory.

The mail server routes all mail from a client. When a client sends mail, the mail server puts it in a queue for delivery. After the mail is in the queue, a user can reboot or turn off the client without losing those mail messages. When the recipient gets mail from a client, the path in the "From " line of the message contains the name of the mail server. If the recipient responds, the response goes to the user's mailbox. Good candidates for mail servers are systems that provide a home directory for users or that are backed up regularly.

If the mail server is not the user's local system, users in configurations using NFS software can mount the `/var/mail` directory by using the `/etc/vfstab` file (if they have root access) or by using the automounter. If NFS support is not available, the users can log in to the server to read their mail.

If users on your network send other types of mail, such as PostScript™ files, audio files, or files from desktop publishing systems, you need to allocate more space on the mail server for mailboxes.

One advantage to establishing a mail server for all mailboxes is that it makes backups easy. Having mail spread over many systems makes it hard to do backups. The disadvantage of storing many mailboxes on one server is that the server can be a single point of failure for many users, but the advantages of providing good backups usually make the risk worthwhile.

Mail Client

A *mail client* is any system that receives mail on a mail server and does not have a local /var/mail directory. This is known as remote mode. Remote mode is enabled by default in /etc/mail/subsidiary.cf.

You must check that the mail client has the appropriate entry in the /etc/vfstab file and a mount point to mount the mailbox from the mail server. Also make sure that the alias for the client is directed to the mail server's host name, not to the client's.

Mail Gateway

The *mail gateway* is a machine that handles connections between networks running different communications protocols or communications between different networks using the same protocol. For example, a mail gateway might connect a TCP/IP network to a network running the Systems Network Architecture (SNA) protocol suite.

The simplest mail gateway to set up is one that connects two networks that use the same protocol or mailer. This system handles mail with an address for which sendmail cannot find a recipient in your domain. If a mail gateway exists, sendmail uses it for sending and receiving mail outside your domain.

You can set up a mail gateway between two networks using unmatched mailers, as shown in Figure 1–2. To support this, you must customize the sendmail.cf file on the mail gateway system, which can be a difficult and time-consuming process.

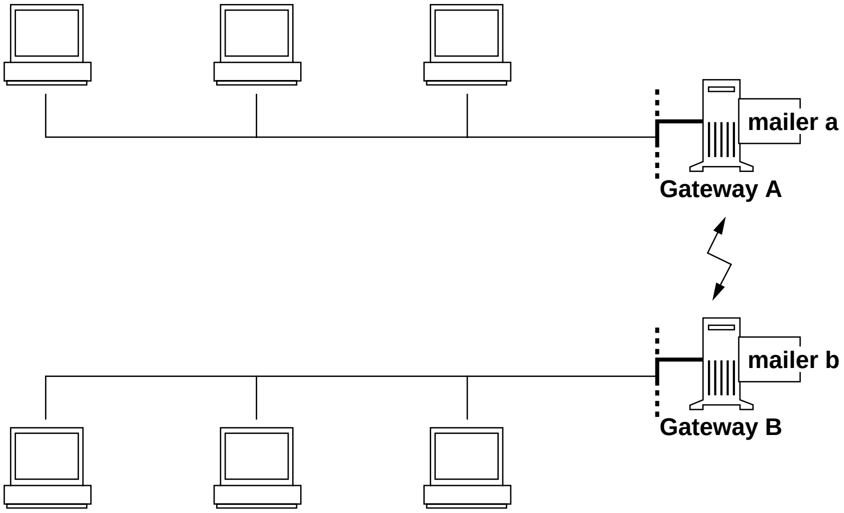

Figure 1–2 Gateway Between Different Communications Protocols

If you have to set up a mail gateway, you should find a gateway-configuration file that is close to what you need and modify it to fit your situation.

If you have a machine providing connections to the Internet, you can configure that machine as the mail gateway. Carefully consider your site's security needs before you configure a mail gateway. You might need to create a firewall gateway between your corporate network and the outside world, and set that up as the mail gateway.

Mail Service Programs and Files

Mail services include many programs and daemons that interact with each other. This section introduces the programs and the terms and concepts related to administering electronic mail. Table 1–3 shows the contents of the `/usr/bin` directory that are used for mail services.

TABLE 1–3 Contents of the `/usr/bin` Directory Used for Mail Services

Name	Type	Description
`aliasadm`	File	A program to manipulate the NIS+ aliases map
`mail`	File	A user agent
`mailcompat`	File	A filter to store mail in SunOS 4.1 mailbox format
`mailq`	Link	Link to `/usr/lib/sendmail`; used to list the mail queue
`mailstats`	File	A program used to read mail statistics stored in the `/etc/mail/sendmail.st` file (if present)
`mailx`	File	A user agent
`mconnect`	File	A program that connects to the mailer for address verification and debugging
`newaliases`	Link	Link to `/usr/lib/sendmail`; used to create the binary form of the aliases file
`rmail`	Link	Link to `/usr/bin/mail`; command often used to permit only the sending of mail
`vacation`	File	A command to set up an automatic reply to mail

Table 1–4 shows the contents of the `/etc/mail` directory.

TABLE 1–4 Contents of the `/etc/mail` Directory

Name	Type	Description
`Mail.rc`	File	Default settings for the `mailtool` user agent
`aliases`	File	Mail–forwarding information
`aliases.dir`	File	Binary form of mail-forwarding information (created by running `newaliases`)
`aliases.pag`	File	Binary form of mail-forwarding information (created by running `newaliases`)
`mailx.rc`	File	Default settings for the `mailx` user agent
`main.cf`	File	Sample configuration file for main systems
`relay-domains`	File	Contains a list of all domains for which relaying is allowed; by default, only the local domain is allowed
`sendmail.cf`	File	Configuration file for mail routing
`sendmail.cw`	File	Optional file that you can create if the number of aliases for the mail host is too long
`sendmail.hf`	File	Help file used by the SMTP `HELP` command
`sendmail.pid`	File	File that lists the PID of the listening daemon
`sendmail.st`	File	The `sendmail` statistics file; if this file is present, `sendmail` logs the amount of traffic through each mailer
`sendmailvars`	File	Stores macro and class definitions for name space lookup from `sendmail.cf`
`subsidiary.cf`	File	Sample configuration file for subsidiary systems

Table 1–5 shows the contents of the `/usr/lib` directory that are used for mail services.

TABLE 1–5 Contents of the /usr/lib Directory Used for Mail Services

Name	Type	Description
mail.local	File	Mailer that delivers mail to mailboxes
sendmail	File	The routing program, also known as the mail transfer agent

Within the /usr/lib directory is a subdirectory that contains all of the files needed to build a sendmail.cf file. The contents of this directory are shown in Table 1–6.

TABLE 1–6 Contents of the /usr/lib/mail Directory Used for Mail Services

Name	Type	Description
README	File	Document describing the configuration files
cf	Directory	Site-dependent and site-independent descriptions of hosts
cf/main-v7sun.mc	File	Main configuration file
cf/makefile	File	Contains rules for building new configuration files
cf/subsidiary-v7sun.mc	File	Configuration file for hosts that NFS-mount /var/mail from another host
domain	Directory	Site-dependent subdomain descriptions
domain/generic.m4	File	Generic domain file from Berkeley
domain/solaris-antispam.m4	File	Domain file with changes that make sendmail function like previous Solaris versions, except that relaying is disabled completely, sender addresses with no hostname are rejected, and unresolvable domains are rejected

TABLE 1–6 Contents of the `/usr/lib/mail` Directory Used for Mail Services *(continued)*

Name	Type	Description
`domain/solaris-generic.m4`	File	Domain file with changes that make `sendmail` function like previous Solaris versions (default)
`feature`	Directory	Definitions of specific features for particular hosts (see `README` for a full description of the features)
`m4`	Directory	Site-independent include files
`mailer`	Directory	Definitions of mailers, which include local, smtp and uucp
`ostype`	Directory	Definitions describing various operating system environments
`ostype/solaris2.m4`	File	Defines local mailer as `mail`
`ostype/solaris2.ml.m4`	File	Defines local mailer as `mail.local` (default)
`sh`	Directory	Shell scripts used by the `m4` build process and migration aids
`sh/check-permissions`	File	Checks permissions of `:include:` aliases and `.forward` files and their parent directory path for correct permissions
`sh/check-hostname`	File	Verifies that `sendmail` is able to determine the fully qualified host name

Several other files and directories are used by the mail services, as shown in Table 1–7.

TABLE 1–7 Other Files Used for Mail Services

Name	Type	Description
`sendmailvars.org_dir`	Table	NIS+ version of `sendmailvars` file
`/etc/shells`	File	Lists the valid login shells
`/usr/sbin/in.comsat`	File	Mail-notification daemon
`/usr/sbin/makemap`	File	Builds binary forms of keyed maps
`/usr/sbin/syslogd`	File	Error message logger, used by `sendmail`
`/usr/dt/bin/dtmail`	File	CDE mail user agent
`/var/mail/mailbox1`, `/var/mail/mailbox2`	File	Mailboxes for delivered mail
`/var/spool/mqueue`	Directory	Storage for undelivered mail
`$OPENWINHOME/bin/mailtool`	File	Window–based mail user agent

Mail services are provided by a combination of these programs, which interact as shown by the simplified diagram in Figure 1–3.

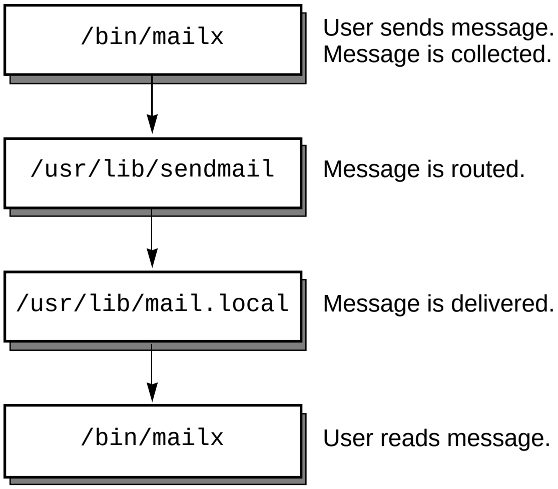

Figure 1–3 How Mail Programs Interact

Users send messages by using programs like mailx or mailtool. See the **mailx**(1) or *mailtool(1)* man pages for information about these programs.

The message is collected by the program that was used to generate it and is passed to the sendmail daemon. The sendmail daemon *parses* the addresses (divides them into identifiable segments) in the message, using information from the configuration file, /etc/mail/sendmail.cf, to determine network name syntax, aliases, forwarding information, and network topology. Using this information, sendmail determines the route a message must take to get to a recipient.

The sendmail daemon passes the message to the appropriate system. The /usr/lib/mail.local program on the local system delivers the mail to the mailbox in the /var/mail/*username* directory of the recipient of the message.

The recipient is notified that mail has arrived, and retrieves it using mail, mailx, mailtool, or a similar program.

sendmail Program

The sendmail program can use different types of communications protocols, like TCP/IP and UUCP. It also implements an SMTP server, message queueing, and mailing lists. Name interpretation is controlled by a pattern-matching system that can handle both domain-based naming and improvised conventions.

The sendmail program can accept domain-based naming as well as arbitrary (older) name syntaxes—resolving ambiguities by using heuristics you specify. sendmail can also convert messages between disparate naming schemes. The domain technique separates the issue of physical versus logical naming. See the *TCP/IP and Data Communications Administration Guide* for a complete description of Internet domain–naming conventions.

You can handle certain special cases by improvised techniques, like providing network names that appear local to hosts on other networks.

The Solaris operating environment uses the sendmail program as a mail router. sendmail is responsible for receiving and delivering electronic mail messages. It is an interface between mail–reading programs like mail, mailx, and mailtool, and mail–transport programs like uucp. The sendmail program controls email messages that users send, evaluates the recipients' addresses, chooses an appropriate delivery program, rewrites the addresses in a format that the delivery agent can handle, reformats the mail headers as required, and finally passes the transformed message to the mail program for delivery.

Note - Solaris releases prior to Solaris 2.4 included a binary called sendmail.mx. This program is now included in the sendmail program and the functionality is turned on by adding the dns flag to the hosts entry in /etc/nsswitch.conf. For more information, see "Setting Up DNS to Work With sendmail" on page 40.

The sendmail program supports three mechanisms for mail rerouting. Which mechanism you choose depends on whether this is a server or domain-wide change, or just a change for one user. In addition, by selecting a different rerouting mechanism, you can change the level of administration required.

One rerouting mechanism is aliasing, which maps names to addresses on a server-wide or a namespace-wide basis, depending on the type of file that is used. Using a namespace alias file allows for mail rerouting changes to be administered at a single source, but there can be lag-times created when the change is propagated. Also, namespace administration is usually restricted to a select group of system administrators, so this is not a change that a normal user is able to make. Rerouting handled through a server alias file is managed by anyone who can become root on that server. Normally, there should be little or no lag-time associated with propagating the change, but the change only affects the local server. This limitation might be acceptable if most of the mail is sent to one server anyway, but trying to propagate this change to many mail servers is easier using a name service. Again, this is not a change that a user is able to administer.

The next mechanisms, forwarding and inclusion, allow users the ability to administer mail rerouting. Forwarding allows local users to reroute their incoming mail to either another mail box, a different mailer, or to another mail host. This form of mail rerouting is supported through the use of .forward files. Further information on these files can be found in ".forward Files" on page 66.

The last rerouting mechanism is inclusion, which allows for alias lists to be maintained by a user instead of requiring root access. To provide this, the root user must create an appropriate entry in the alias file on the server. After this entry is created, then the user can reroute mail as needed. More information on inclusion can be found in "/etc/mail/aliases" on page 64.

Figure 1–4 shows how sendmail uses aliases. Programs that read mail, like /usr/bin/mailx, can have aliases of their own, which are expanded before the message reaches sendmail. The aliases for sendmail can come from a number of name space sources (local files, NIS or NIS+). The order of the lookup is determined by the nsswitch.conf file. See the **nsswitch.conf**(4) man page.

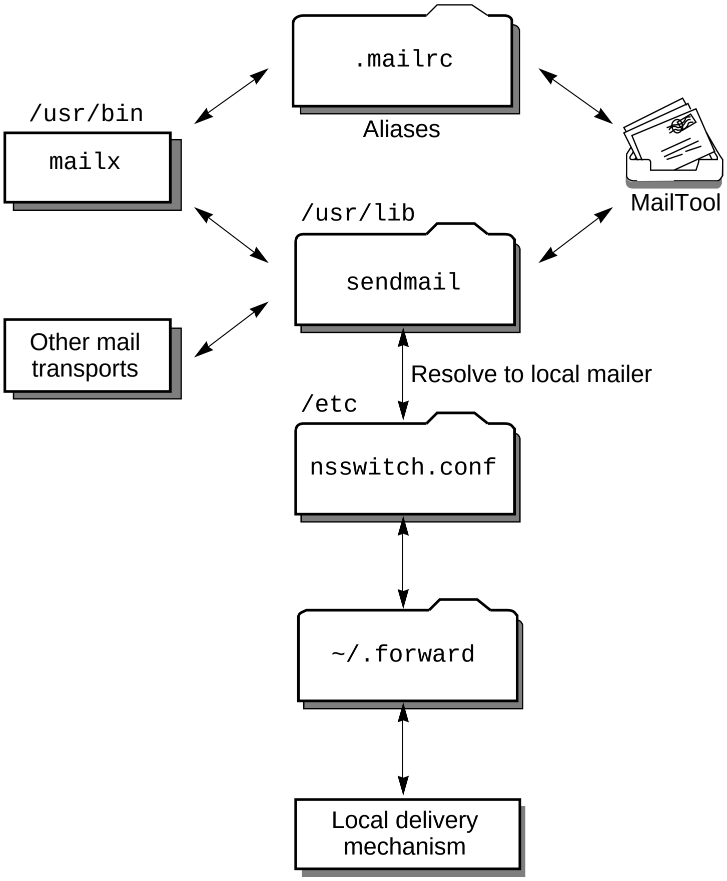

Figure 1–4 How `sendmail` Uses Aliases

`sendmail` Features

The `sendmail` program provides the following features:

- It is reliable. It is designed to correctly deliver every message. No message should ever be completely lost.
- It uses existing software for delivery whenever possible.
- It can be configured to handle complex environments, including multiple connections to a single network type (like with UUCP or Ethernet). `sendmail` checks the contents of an address as well as its syntax to determine which mailer to use.
- It uses configuration files to control mail configuration instead of requiring that configuration information is compiled into the code.

- Users can maintain their own mailing lists. In addition, individuals can specify their own forwarding without modifying the domain-wide alias file (typically located in the domain-wide aliases maintained by NIS or NIS+).
- Each user can specify a custom mailer to process incoming mail, which can provide functions like returning an "I am on vacation" message. See the **vacation**(1) man page for more information.
- It batches addresses to a single host to reduce network traffic.

Figure 1–5 shows how sendmail interacts with the other programs in the mail system.

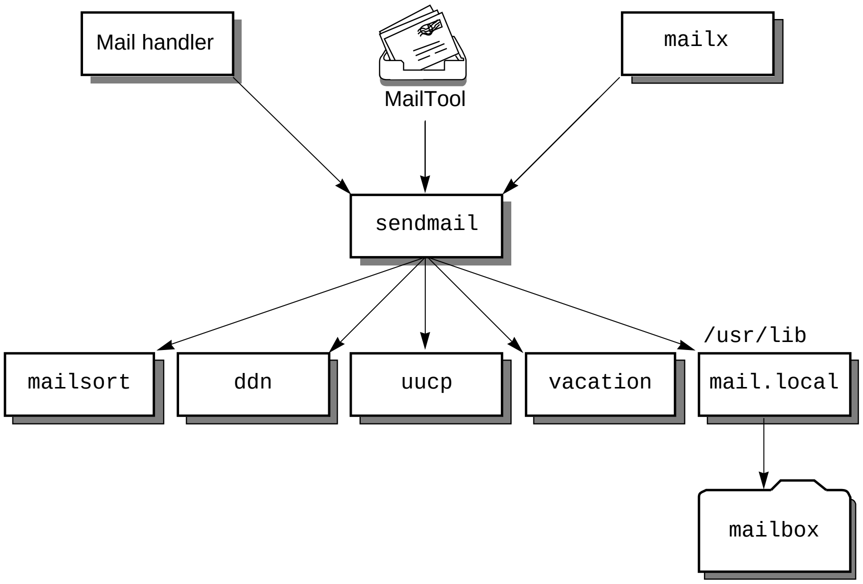

Figure 1–5 Interaction of sendmail With Other Mail Programs

The user interacts with a mail-generating and -sending program. When the mail is submitted, the mail-generating program calls sendmail, which routes the message to the correct mailers. Because some of the senders might be network servers and some of the mailers might be network clients, sendmail can be used as an Internet mail gateway.

sendmail Configuration File

A *configuration file* controls the way that sendmail performs its functions. The configuration file determines the choice of delivery agents, address rewriting rules, and the format of the mail header.

The sendmail program uses the information from the /etc/mail/sendmail.cf file to perform its functions. Each system has a default sendmail.cf file installed in the /etc/mail directory. You do not need to edit or change the default configuration file for mail servers or mail clients. The only systems that require a customized configuration file are mail hosts and mail gateways.

The Solaris operating environment provides two default configuration files in the /etc/mail directory:

1. A configuration file named main.cf for the system (or systems) you designate as the mail host or a mail gateway
2. A configuration file named subsidiary.cf (a duplicate copy of the default sendmail.cf file)

The configuration file you use on a system depends on the role the system plays in your mail service.

- For mail clients or mail servers, you do not need to do anything to set up or edit the default configuration file.
- To set up a mail host or gateway, copy the main.cf file and rename it sendmail.cf (in the /etc/mail directory). Then reconfigure the sendmail configuration file to set the relay mailer and relay host parameters needed for your mail configuration.

The following list describes some configuration parameters you might want to change, depending on the requirements of your site:

- Time values specifies:
  - Read timeouts
  - Length of time a message remains undelivered in the queue before it is returned to the sender
- Delivery modes specify how quickly mail will be delivered.
- Load limiting prevents wasted time during loaded periods by not attempting to deliver large messages, messages to many recipients, and messages to sites that have been down for a long time.
- Log level specifies what kinds of problems are logged.

CHAPTER **2**

Setting Up and Administering Mail Services

This chapter describes how to set up and administer mail services. If you are not familiar with administering mail services, read Chapter 1, for an introduction to the terminology and structure of the mail services and for descriptions of several mail service configurations.

Use the following table to find the page for specific task instructions:

Planning Your Mail System

This section describes four basic types of mail configurations and briefly outlines the tasks required to set up each configuration. You might find this section useful if you need to set up a new mail system or if you are expanding an existing one. The configurations start with the most basic case (mail completely local, no connection to the outside world) and increase in complexity to a two-domain configuration with a mail gateway.

To set up a mail system, regardless of its configuration, you need these elements:

- A `sendmail.cf` configuration file on each system
- Alias files with an alias for each user to point to the place where mail is stored
- A mailbox to store (or spool) mail files for each user
- A `postmaster` alias for the person who administers mail services

How you set up the configuration file and the alias file and where you put the mailboxes depend on the configuration you choose.

As system administrator, you should decide on a policy for updating aliases and for forwarding mail messages. You might set up an `aliases` mailbox as a place for users to send requests for mail forwarding and for changes to their default mail alias. If your system uses NIS or NIS+, you can administer forwarding rather than forcing users to manage it themselves.

Local Mail Only

The simplest mail configuration, shown in Figure 2–1, is one mail host with two or more workstations connected to it. Mail is completely local. All the clients store mail on their local disks and are acting as mail servers. Mail addresses are parsed using the `/etc/mail/aliases` files.

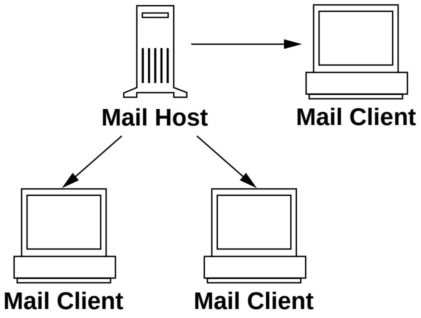

Figure 2–1 Local Mail Configuration

To set up this kind of mail configuration, you need:

- The default `/etc/mail/sendmail.cf` file on each mail client system (no editing required)
- A server designated as the mail host (add `mailhost.`*domainname* to the `/etc/hosts` file on the mail host; then if you are not running NIS or NIS+, add the mail host IP address line to the `/etc/hosts` file of all mail clients)
- Matching `/etc/mail/aliases` files on any system that has a local mailbox (unless you are running NIS or NIS+)
- Enough space in `/var/mail` on each mail client system to hold the mailboxes

Local Mail in Remote Mode

In this configuration, each mail client mounts its mail from one mail server that provides mail spooling for client mailboxes. This server can also be the mail host. This configuration makes it easy to back up the mailboxes for each client.

To set up this kind of mail configuration, you need:

- The default `/etc/mail/sendmail.cf` file on each mail client system (no editing required)
- A server designated as the mail host (add `mailhost.`*domainname* to the `/etc/hosts` file on the mail host; then if you are not running NIS or NIS+, add the mail host IP address line to the `/etc/hosts` file of all mail clients)
- Matching `/etc/mail/aliases` files on any system that has a local mailbox (unless you are running NIS or NIS+)
- Entries in each mail client's `/etc/vfstab` file or `/etc/auto_direct` (if autofs is used) to mount the `/var/mail` directory
- Enough space in `/var/mail` on the mail server to hold the client mailboxes

Local Mail and a Remote Connection

The most common mail configuration in a small network is shown in Figure 2–2. One system is the mail server, the mail host, and the mail gateway to the outside world. Mail is distributed using the `/etc/mail/aliases` files. No name service is required.

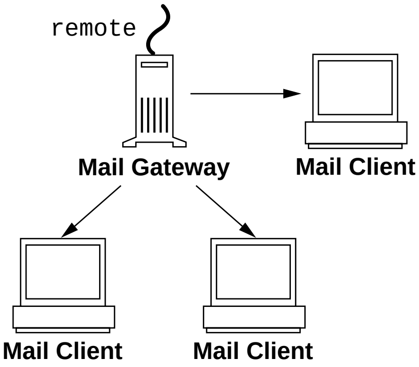

Figure 2–2 Local Mail Configuration With a UUCP Connection

To set up this kind of a mail configuration, assuming that the mail clients mount their mail files from `/var/mail` on the mail host, you need:

- The `main.cf` file on the mail gateway (no editing required if MX records are used)

- The default /etc/mail/sendmail.cf file on each mail client system (no editing required)
- A server designated as the mail host (add mailhost.*domainname* to the /etc/hosts file on the mail host; if you are not running NIS or NIS+, add the mail host IP address line to the /etc/hosts file of all mail clients)
- Matching /etc/mail/aliases files on any system that has a local mailbox (unless you are running NIS or NIS+)
- Entries in each mail client's /etc/vfstab file or /etc/auto_direct (if autofs is used) to mount the /var/mail directory when mailboxes are located on the mail host
- Enough space in /var/mail on the mail server to hold the client mailboxes

Two Domains and a Gateway

The mail configuration shown in Figure 2–3 has two domains and a mail gateway. In this configuration, the mail server, the mail host, and the mail gateway (or gateways) for each domain are likely to be different systems. To make the process of administering and distributing mail easier, a name service is used.

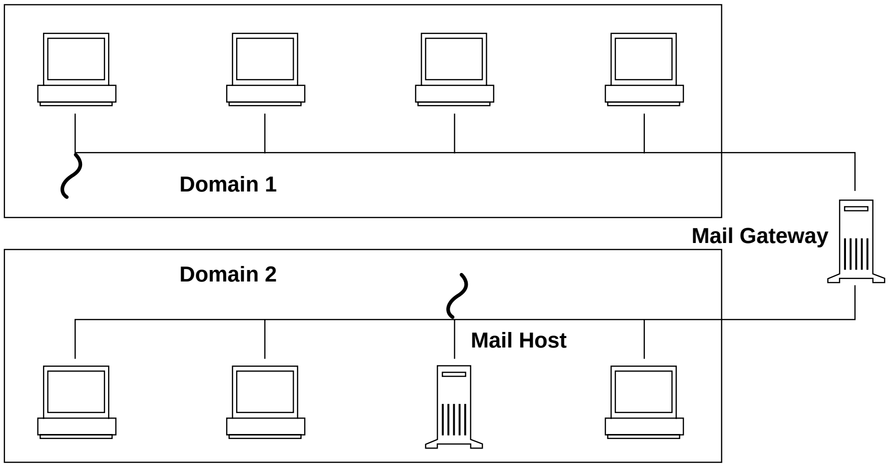

Figure 2–3 Two Domains and a Gateway

To set up this kind of a mail configuration, assuming that the mail clients mount their mail files from /var/mail on the mail host, you need:

- Complex gateway systems requiring a customized sendmail.cf file with special rules added
- The main.cf file on the mail gateway (no editing required if you use MX records)

- A server designated as the mail host (add `mailhost.`*domainname* to the `/etc/hosts` file on the mail host; if you are not running NIS or NIS+, add the mail host IP address line to the `/etc/hosts` file of all mail clients)
- Matching `/etc/mail/aliases` files on any system that has a local mailbox (unless you are running NIS or NIS+)
- An alias entry for each user, to point to where the mail is stored, in `mail_aliases.org_dir` for NIS+ or the aliases map for NIS
- The default `/etc/mail/sendmail.cf` file on each mail client system (no editing required)
- Entries in each mail client's `/etc/vfstab` file or `/etc/auto_direct` (if autofs is used) to mount the `/var/mail` directory when mailboxes are located on the mail host
- Enough space in `/var/mail` on the mail server to hold the client mailboxes

Setting Up Mail Services

You can set up a mail service relatively easily if your site does not provide connections to electronic mail (email) services outside your company or if your company is in a single domain.

Mail requires two types of configurations for local mail and two more for communication with networks outside of your domain. You can combine these configurations on the same system or provide them on separate systems. You need to set up systems on your site to perform the functions described in Table 2–1.

TABLE 2–1 Mail Configurations

Configuration	**Description**
Mail client	Mail clients are users who have mailboxes on a mail server.
Mail server	The mail server stores mailboxes in the `/var/mail` directory.
Mail host	You need at least one mail host. The mail host resolves difficult email addresses and reroutes mail within your domain.
Mail gateway	A mail gateway is a connection between different networks outside your domain or between differing communications networks. If you have to set up a mail gateway, you should find a gateway configuration file that is close to what you need and modify it to fit your situation.

Before you begin to set up your mail service, choose the systems to act as mail servers, mail hosts, and mail gateways. You should also make a list of all the mail clients for which you are providing service and include the location of their mailboxes. This list will help you when you are ready to create mail aliases for your users. See Chapter 1, for more information about the function each of these systems provides. For your convenience, guidelines about which systems are good candidates for mail server, mail host, and mail gateways are repeated in the following sections.

To simplify the setup instructions, this chapter tells you what you need to do to set up individual mail servers, mail hosts, mail clients, and relay hosts. If a system in your mail services configuration is acting in more than one capacity, follow the appropriate instructions for each type of system. For example, if your mail host and mail server functions are on the same system, follow the directions for setting up that system as a mail host and then follow the directions for setting up the same system as a mail server.

Note - The following procedures for setting up a mail server and mail client apply when mailboxes are NFS-mounted. However, mailboxes typically are maintained in locally mounted `/var/mail` directories—in which case the following procedures are not needed.

▼ How to Set Up a Mail Server

No special steps are required to set up a mail server that is only serving mail for local users. The user must have an entry in the password file or in the name space, and the user should have a local home directory (so that `~/.forward` can be checked) for mail to be delivered. This is why home directory servers are often set up as the mail server.

The mail server can route all mail for many mail clients. The only resource requirement for this type of mail server is that it have adequate spooling space for client mailboxes. The `/var/mail` directory must be made available for remote mounting.

For this task, check the `/etc/dfs/dfstab` file to be sure the `/var` directory is exported.

1. **Type** `share` **and press Return.**

   If the `/var` directory is shared, you do not need to do more. If the `/var` directory is not exported, continue with the next step.

2. **Type** `share -F nfs /var/mail` **and press Return.**

3. **To permanently share the file system, edit** `/etc/dfs/dfstab` **and add the command line used in step 2.**

Note - The `mail.local` program automatically creates mailboxes in the `/var/mail` directory the first time a message is delivered. You do not need to create individual mailboxes for your mail clients.

▼ How to Set Up a Mail Client

A mail client is a user of mail services, with a mailbox on a mail server, and a mail alias in the `/etc/mail/aliases` file that points to the location of the mailbox.

1. **Become root on the mail client's system.**

2. **Make sure that there is a** `/var/mail` **mount point on the mail client's system.**

3. **Mount the** `/var/mail` **directory from the mail server.**

   The mail directory can be automatically mounted or mounted at boot time.

   a. **To mount** `/var/mail` **automatically, edit** `/etc/auto_direct` **and add an entry like this one:**

   ```
   /var/mail -rw,hard,actimeo=0 server:/var/mail
   ```

   b. **To mount** `/var/mail` **at boot time, edit the** `/etc/vfstab` **file and add an entry for the** `/var/mail` **directory on the mail server, mounting it on the local** `/var/mail` **directory.**

   ```
   server:/var/mail - /var/mail nfs - no rw,hard,actimeo=0
   ```

   The client's mailbox is automatically mounted any time the system is rebooted. Type `mountall` to mount the client mailbox until the system is rebooted.

Caution - You must include the `actimeo=0` option when mounting mail from an NFS server to allow mailbox locking and access to work properly.

4. **Use the Administration Tool to edit the** `/etc/hosts` **file and add an entry for the mail server.**

   This step is not required if you are using a name service.

5. **Add an entry for the client to one of the alias files.**

   See "Creating Mail Aliases" on page 33 for information about how to create mail aliases for different kinds of mail configurations.

Note - The `mail.local` program automatically creates mailboxes in the `/var/mail` directory the first time a message is delivered. You do not need to create individual mailboxes for your mail clients.

How to Set Up a Mail Host

A mail host resolves email addresses and reroutes mail within your domain. A good candidate for a mail host is a system that connects your systems to the outside world or to a parent domain.

1. **Become root on the mail host system.**

2. **Verify the hostname configuration.**

   Run the `check-hostname` script to verify if `sendmail` will be able to identify the fully qualified hostname for this server:

```
% /usr/lib/mail/sh/check-hostname
hostname phoenix OK: fully qualified as phoenix.eng.acme.com
```

   If this script is not successful in identifying the fully qualified hostname, you need to add the fully qualified hostname as the first alias for the host in `/etc/hosts`.

3. **Use the Administration Tool to edit the `/etc/hosts` file.**

   Add the word `mailhost` and `mailhost.`*domainname* after the IP address and system name of the mail host system. The system is designated as a mail host. The *domainname* should be identical to the string given as the subdomainname in the output of this command: `/usr/lib/sendmail -bt -d0 </dev/null`.

```
% /usr/lib/sendmail -bt -d0 </dev/null
Version 8.9.0+Sun
 Compiled with: MAP_REGEX LOG MATCHGECOS MIME7TO8 MIME8TO7 NAMED_BIND
                NDBM NETINET NETUNIX NEWDB NIS NISPLUS QUEUE SCANF SMTP
                USERDB XDEBUG

============ SYSTEM IDENTITY (after readcf) ============
      (short domain name) $w = phoenix
  (canonical domain name) $j = phoenix.eng.acme.com
         (subdomain name) $m = eng.acme.com
              (node name) $k = phoenix
```

(continued)

(Continuation)

```

=======================================================
```

4. **Create an entry for the new mail host in one of the hosts files.**

   If you are using NIS or NIS+, add an entry including a host alias called `mailhost` and `mailhost.`*domainname* to the host entry for the new mail host.

   If you are not using NIS or NIS+, you must create an entry in `/etc/hosts` for each system on the network. The entry should use this format:
   IP_address mailhost_name `mailhost` `mailhost.`*domainname*

5. **Type** `cp /etc/mail/main.cf /etc/mail/sendmail.cf` **and press Return.**

   This copies and renames the `/etc/mail/main.cf` file.

6. **Restart** `sendmail` **and test your mail configuration.**

   See “How to Test the Mail Configuration” on page 45 for information.

▼ How to Set Up a Mail Gateway

A mail gateway manages communication with networks outside of your domain. The mailer on the sending mail gateway can match the mailer on the receiving system.

A good candidate for a mail gateway is a system attached to Ethernet and phone lines or a system configured as a router to the Internet. You might want to configure the mail host or another system as mail gateway. You might choose to configure more than one mail gateway for your domain. If you have UUCP connections, you should configure the system (or systems) with UUCP connections as the mail gateway.

1. **Become root on the mail gateway.**

2. **Type** `cp /etc/mail/main.cf /etc/mail/sendmail.cf` **and press Return.**

   This command copies and renames the `main.cf` file.

3. **Verify the hostname configuration.**

   Run the `check-hostname` script to verify if `sendmail` will be able to identify the fully qualified hostname for this server:

```
% /usr/lib/mail/sh/check-hostname
hostname phoenix OK: fully qualified as phoenix.eng.acme.com
```

If this script is not successful in identifying the fully qualified hostname, you need to add the fully qualified hostname as the first alias for the host in `/etc/hosts`.

4. **Restart `sendmail` and test your mail configuration.**

   See "How to Test the Mail Configuration" on page 45 for information.

Building a `sendmail` Configuration File

The process to create `sendmail` configuration files has been changed. For many sites, adminstration of the configuration files should now be easier. Although it is still acceptable to use older version of `sendmail.cf` files, it would be best to move to the new system as soon as is reasonable. A complete description of the new process is described in `/usr/lib/mail/README`.

▼ How to Build a New `sendmail.cf` File

1. **Make a copy of the configuration files that you want to change.**

```
# cd /usr/lib/mail/cf
# cp main-v7sun.mc myhost.mc
```

2. **Edit the new configuration files as needed (for example *myhost*`.mc`).**

3. **Build the configuration file using `m4`.**

```
# cd /usr/lib/mail/cf
# /usr/ccs/bin/make myhost.cf
```

4. **Test the new configuration file using the –C option to specify the new file.**

```
# /usr/lib/sendmail -C /usr/lib/mail/cf/myhost.cf -v testaddr </dev/null
```

This command sends a message to `testaddr` while displaying messages as it runs. Only outgoing mail can be tested without restarting the `sendmail` service on the system. For systems that are not handling mail yet, use the full testing procedure found in “How to Test the Mail Configuration” on page 45.

5. **Install the new configuration file after making a copy of the original.**

```
# cp /etc/mail/sendmail.cf /etc/mail/sendmail.cf.save
# cp /usr/lib/mail/cf/myhost.cf /etc/mail/sendmail.cf
```

6. **Restart the sendmail service.**

```
# pkill -HUP sendmail
```

Creating Mail Aliases

You can use the `aliasadm` command to create mail aliases for a user. Mail aliases must be unique within the domain. This section tells you how to use command lines to search the mail aliases table for aliases, and to create mail aliases for NIS+, NIS, DNS, or on the local system. Or you can use the Administration Tool’s Database Manager application to perform these tasks on the aliases database.

In addition, database files can be created for the local mail host using `makemap`. Using these database files, does not provide the all of the advantages of using a name space like NIS or NIS+, but retrieving the data should be faster than using local files.

▼ How to List the Contents of an NIS+ Aliases Table

To use the `aliasadm` command, you must be either `root`, a member of the NIS+ group that owns the `mail_aliases` table, or the person who created the table.

To List the Entire Contents of the NIS+ `mail_aliases` Table

♦ **Type** `aliasadm -l` **and press Return.**

This lists the contents of the aliases table in alphabetical order by alias.

Note - If you have a large aliases table, listing the entire contents can take some time. If you are searching for a specific entry, pipe the output through the `grep` command (`aliasadm -l | grep` *entry*) so that you can use the `grep` search capability to find specific entries.

To List Individual Entries in the NIS+ `mail_aliases` Table

♦ **Type** `aliasadm -m` ***alias*** **and press Return.**

The alias entry is listed.

```
# aliasadm -m ignatz
ignatz: ignatz@saturn # Alias for Iggy Ignatz
```

Note - The `aliasadm -m` option matches only the complete alias name. It does not match partial strings. You cannot use metacharacters (like `*` and `?`) with the `aliasadm -m` option. If you are interested in partial matches, type `aliasadm -l | grep` *partial-string* and press Return.

▼ How to Add Aliases to a NIS+ `mail_aliases` Table From the Command Line

If you are creating a completely new NIS+ `mail_aliases` table, you must first initiate the NIS+ table.

To Initiate a NIS+ Table

♦ **Type** `aliasadm -I` **and press Return.**

To Add Aliases to a NIS+ `mail_aliases` Table From the Command Line

1. **Compile a list of each of your mail clients, the locations of their mailboxes, and the names of the mail server systems.**

2. **Become root on any system.**

3. **For each alias, type** `aliasadm -a` ***alias expanded_alias*** [***options comments***] **and press Return.**

   This adds the aliases to the NIS+ aliases table.

   ```
   # aliasadm -a iggy iggy.ignatz@saturn "Iggy Ignatz"
   ```

4. **Type** `aliasadm -m` ***alias*** **and press Return.**

   This displays the entry you created.

5. **Check that the entry is correct.**

▼ How to Add Entries by Editing a NIS+ `mail_aliases` Table

If you are adding more than two or three aliases, you might want to edit the NIS+ table directly.

1. **Compile a list of each of your mail clients, the locations of their mailboxes, and the names of the mail server systems.**

2. **Become root on any system.**

3. **Type** `aliasadm -e` **and press Return.**

   The aliases table is displayed using the editor set with the `$EDITOR` environment variable. If the variable is not set, `vi` is the default editor.

4. **Type each alias on a separate line, using these formats:**

   a. **Type the aliases in any order, at any place in the table.**

      The order is not important to the NIS+ mail_aliases table. The `aliasadm -l` command sorts the list and displays them in alphabetical order.

b. **Use the format *alias*: *expanded_alias* # ["*option*"# "*comments*"]**

   If you leave the option column blank, type an empty pair of quotation marks ("") and then add the comments.

c. **End each line by pressing Return.**

5. **Check that the entries are correct.**

6. **Save the changes.**

▼ How to Change Entries in a NIS+ `mail_aliases` Table

1. **Become root on any system.**

2. **Type** `aliasadm -m` ***alias*** **and press Return.**

   The information for the alias is displayed.

3. **Type** `aliasadm -c` ***alias expanded_alias*** [***options comments***] **and press Return.**

   The alias is changed using the new information you provide.

4. **Type** `aliasadm -m` ***alias*** **and press Return.**

   The entry you created is displayed.

5. **Check that the entry is correct.**

▼ How to Delete Entries From a NIS+ `mail_aliases` Table

1. **Become root on any system.**

2. **Type** `aliasadm -d` ***alias*** **and press Return.**

   The alias is deleted from the NIS+ `mail_aliases` table.

Setting Up NIS `mail.aliases` Map

The `/etc/mail/aliases` file on an NIS master contains all names by which a system or person is known. The NIS master is searched if there is no match in the local `/etc/mail/aliases` files. The `sendmail` program uses the NIS master file to determine mailing addresses. See the **`aliases`**(4) man page.

You can either edit the file on each system or edit the file on one system and copy it to each of the other systems.

Aliases are in the following form:

```
name:  name1,  name2,  ...
```

You can use aliases for local names or domains. For example, an alias entry for user `fred` who has a mailbox on the system `saturn` and who is in the domain `planets` would have this entry in the `/etc/mail/aliases` file:

```
fred: fred@planets
```

How to Set Up NIS `mail.aliases` Map

1. **Compile a list of each of your mail clients, the locations of their mailboxes, and the names of the mail server systems.**

2. **Become root on the NIS master server.**

3. **Edit the** `/etc/mail/aliases` **file, and make the following entries:**

   a. **Add an entry for each mail client.**

   b. **Change the entry** `Postmaster: root` **to the mail address of the person who is designated as postmaster.**

      See “Setting Up the Postmaster Alias” on page 40 for more information.

   c. **If you have created a mailbox for administration of a mail server, create an entry for** `root:`***mailbox@mailserver.***

   d. **Save the changes.**

4. **Edit the** `/etc/hosts` **file on the NIS master server and create an entry for each mail server.**

5. **Type `cd /var/yp` and press Return.**

6. **Type `make` and press Return.**

The changes in the /etc/hosts and /etc/mail/aliases files are propagated to NIS slave systems. It takes a few minutes, at most, for the aliases to take effect.

Setting Up Local Mail Aliases Files

The /etc/mail/aliases file on a local system contains all names by which a system or person is known. The sendmail program uses this file to determine mailing addresses. See the **aliases**(4) man page.

If your network is not running a name service, the /etc/mail/aliases file of each system should contain entries for all mail clients. You can either edit the file on each system or edit the file on one system and copy it to each of the other systems.

Aliases are of the form:

```
name:  name1,  name2,  ...
```

You can create aliases for only local names—a current host name or no host name. For example, an alias entry for user ignatz who has a mailbox on the system saturn would have this entry in the /etc/mail/aliases file:

```
ignatz: ignatz@saturn
```

It is a good idea to create an administrative account for each mail server. You do this by assigning root a mailbox on the mail server and adding an entry to the /etc/mail/aliases file for root. For example, if the system saturn is a mailbox server, add the entry root: sysadmin@saturn to the /etc/mail/aliases file.

How to Set Up Local Mail Aliases Files

1. **Compile a list of each of your mail clients and the locations of their mailboxes.**

2. **Become root on the mail server.**

3. **Edit the /etc/mail/aliases file and make the following entries:**

   a. **Add an entry for each mail client.**

   b. **Change the entry Postmaster: root to the mail address of the person who is designated as postmaster.**

      See “Setting Up the Postmaster Alias” on page 40 for more information.

   c. **If you have created a mailbox for administration of a mail server, create an entry for root: *mailbox@mailserver*.**

   d. **Save the changes.**

4. **Type** `newaliases` **and press Return.**

   This creates an `alias` file in binary form that `sendmail` can use. The file is stored in the `/etc/mail/aliases.dir` and `/etc/mail/aliases.pag` files.

5. **Copy the** `/etc/mail/aliases`**, the** `/etc/mail/aliases.dir`**, and** `/etc/mail/aliases.pag` **files to each of the other systems.**

   When you copy all three files you do not need to run the `newaliases` command on each of the other systems.

   You can copy the files by using the `rcp` or `rdist` commands or by using a script that you create for this purpose. Remember that you must update all the `/etc/mail/aliases` files each time you add or remove a mail client.

How to Create a Keyed Map File

1. **Using the editor of your choice, create the input file.**

   Entries can look like the following:

```
sandy@newdomain.com     ssmith@newdomain.com
ssmith@olddomain.com    error:nouser No such user here
@olddomain.com          %1@newdomain.com
```

   In this sample, the first entry redirects mail to a new alias; the second entry creates a message when an incorrect alias is used; and the last entry redirects all incoming mail from `olddomain` to `newdomain`.

2. **Make the database file.**

```
# /usr/sbin/makemap -o dbm newmap < newmap
```

–o

Append to instead of overwritting the file. See **`makemap`**(1M) for a list of the options available.

dbm

Selects the `dbm` database type. Other map types are `btree` or `hash`.

newmap

Is the name of the input file and the first part of the name of the database file. If the `dbm` database type is selected, then database files are created using a `.pag` and a `.dir` suffix. For the other two database types, the file name is followed by `.db`.

Setting Up DNS to Work With `sendmail`

The DNS name service does not support aliases for individuals. It does support aliases for hosts or domains using *mail exchange* (MX) *records* and cname records. You can specify host names, domain names, or both in the DNS database. See the *Solaris Naming Setup and Configuration Guide* for more information about administering DNS.

How to Use DNS With `sendmail`

1. **Edit the `/etc/nsswitch.conf` file and make sure that the hosts entry includes the `dns` flag.**

   The host entry must include the `dns` flag for the DNS host aliases to be used.

2. **Check for a `mailhost` and `mailhost.`*domainname* entry.**

   Make sure there is an entry for `mailhost` and `mailhost.`*domainname* in the DNS database.

Setting Up the Postmaster Alias

Every system should be able to send mail to a `postmaster` mailbox. You can create a NIS or NIS+ alias for `postmaster` or create one in each local `/etc/mail/aliases` file. Here is the default `/etc/mail/aliases` entry:

```
# Following alias is required by the mail protocol, RFC 822
# Set it to the address of a HUMAN who deals with this system’s
# mail problems.
Postmaster: root
```

To create the `postmaster` alias, edit each system’s `/etc/mail/aliases` file and change `root` to the mail address of the person who acts as postmaster.

You might want to create a separate mailbox for the postmaster to keep postmaster mail separate from personal mail. If you create a separate mailbox, use the mailbox address instead of the postmaster’s mail address when you edit the `/etc/mail/aliases` files.

How to Create a Separate Mailbox for postmaster

1. **Create a user account for the person designated as** postmaster **and put an asterisk (*) in the password field.**

2. **After mail has been delivered, type** `mail -f` ***postmaster*** **and press Return.**

   The `mail` program will be able to read and write to the mailbox name.

How to Add the postmaster Mailbox to the Aliases

1. **Become root and edit the** `/etc/mail/aliases` **file on each system.**

   If your network does not run NIS or NIS+, edit the `/etc/mail/aliases` file.

2. **Change the** `postmaster` **alias from** `root` **to** `Postmaster:` ***postmastermailbox@postmasterhost*** **and save the changes.**

3. **On the postmaster's local system create an entry in the** `/etc/mail/aliases` **file that defines the name of the alias (**`sysadmin`**, for example) and includes the path to the local mailbox.**

4. **Type** `newaliases` **and press Return.**

   Or you could change the `Postmaster:` entry in the `aliases` file to a `Postmaster`: */usr/somewhere/somefile* entry.

Administering the Mail Queue

This section describes how to keep the mail service running smoothly.

▼ How to Print the Queue

You can print the contents of the queue with `mailq`.This command is equivalent to specifying the `-bp` flag to `sendmail`.

♦ **Type** `/usr/bin/mailq | more` **and press Return.**

A list of the queue IDs, the size of the message, the date the message entered the queue, the message status, and the sender and recipients are displayed.

▼ How to Force the Queue

♦ **Type `/usr/lib/sendmail -q -v` and press Return.**

This forces the processing of the queue and displays progress of the jobs as the queue is cleared.

▼ How to Run a Subset of the Mail Queue

♦ **Type `/usr/lib/sendmail -qR`*string* and press Return.**

You can run a subset of the queue at any time with the `-qR`*string* (run queue where any recipient name matches *string*) or with `-qI`*nnnnn* (run just one message with queue ID *nnnnn*). The *string* can also match host names, so any sub-string of *user@host.domain* will match.

This example processes everything in the queue for recipient `wnj`.

```
# /usr/lib/sendmail -qRwnj
```

▼ How to Move the Queue

1. **Become root on the mail host.**

2. **Type `/etc/init.d/sendmail stop` and press Return.**

   This kills the old `sendmail` daemon to keep it from trying to process the old queue directory.

3. **Type `cd /var/spool` and press Return.**

4. **Type `mv mqueue omqueue; mkdir mqueue` and press Return.**

   This moves the directory, `mqueue`, and all its contents to the `omqueue` directory and then creates a new empty `Rmqueue` directory.

5. **Type `chmod 755 mqueue; chown daemon.daemon mqueue` and press Return.**

   These commands set the permissions of the directory to read/write/execute by owner, and read/execute by group and others; these commands also set the owner and group to `daemon`.

6. **Type `/etc/init.d/sendmail start` and press Return.**

   This starts a new `sendmail` daemon.

▼ How to Run the Old Mail Queue

1. **Type `/usr/lib/sendmail -oQ/var/spool/omqueue -q` and press Return.**

   The `-oQ` flag specifies an alternate queue directory and the `-q` flag says to run every job in the queue. Use the `-v` flag if you want to see the verbose output displayed on the screen.

2. **When the queue is finally emptied type `rmdir /var/spool/omqueue` and press Return.**

   This removes the empty directory.

Administering `.forward` Files

This section contains several procedures related to `.forward` file administration. Because these files can be edited by users, the files can cause problems.

▼ How to Disable `.forward` Files

This procedure disables `.forward` files only for a particular host.

1. **Become root.**

2. **Make a copy of `/usr/lib/mail/domain/solaris-generic.m4` or your site-specific domain m4 file:**

```
# cd /usr/lib/mail/domain
# cp solaris-generic.m4 myhost.domain.m4
```

3. **Add the following line to the file you just created:**

```
define('confFORWARD_PATH','')dnl
```

   If a line already exists in the domain m4 file that you are using, replace the line.

4. **Build and install a new configuration file.**

   See “How to Build a New `sendmail.cf` File” on page 32 for a complete procedure.

How to Change the .forward File Search Path

1. **Become root.**

2. **Make a copy of** /usr/lib/mail/domain/solaris-generic.m4 **or your site-specific domain m4 file:**

```
# cd /usr/lib/mail/domain
# cp solaris-generic.m4 myhost.domain.m4
```

3. **Add a line like the following to the file you just created:**

```
define(‘confFORWARD_PATH’,‘~z/.forward:/var/forward/$u’)dnl
```

   If a line already exists in the domain m4 file that you are using, replace the line.

4. **Build and install a new configuration file.**

   See “How to Build a New sendmail.cf File” on page 32 for a complete procedure.

How to Create and Populate /etc/shells

This file is not included in the standard release, so it must be added if users are to be allowed to use .forward files to forward mail to a program or to a file. It is possible to create the file by hand by using grep to identify all of the shells listed in your password file, then entering them manually in the file, but it is easier to use the procedure below, which employes a script that can be downloaded.

1. **Download the script from** http://www.sendmail.org/sun-specific/gen-etc-shells.html**.**

2. **Become root.**

3. **To generate a list of shells, run the** gen-etc-shells **script.**

```
# ./gen-etc-shells.sh > /tmp/shells
```

   This script uses the getent command to collect the names of shells included in the password file sources listed in /etc/nsswitch.conf.

4. **Inspect the list of shells in `/tmp/shells`.**

   Using the editor of your choice, remove any shells that you do not want included.

5. **Move the file to `/etc/shells`.**

```
# mv /tmp/shells /etc/shells
```

Troubleshooting Tips

This section provides some tips and tools that you can use for troubleshooting problems with the mail services.

▼ How to Test the Mail Configuration

1. **Restart sendmail on any system for which you have changed a configuration file.**

```
# pkill -HUP sendmail
```

2. **Send test messages from each system by typing `/usr/lib/sendmail -v` *names* `</dev/null` and press Return.**

   Specify a recipient’s email address in place of the *names* variable.

   This command sends a null message to the specified recipient and displays messages while it runs.

3. **Run these tests:**

   a. **Send mail to yourself or other people on the local system by addressing the message to a regular user name.**

   b. **If you are on Ethernet, send mail to someone on another system.**

      Do this in three directions: from the main system to a client system, from a client system to the main system, and from a client system to another client system.

   c. **If you have a mail gateway, send mail to another domain from the mail host to ensure that the relay mailer and host are configured properly.**

d. **If you have set up a UUCP connection on your phone line to another host, send mail to someone at that host and have that person send mail back or call you when the message is received.**

e. **Ask someone to send mail to you over the UUCP connection.**

The `sendmail` program cannot tell whether the message gets through, because it hands the message to UUCP for delivery.

f. **Send a message to `postmaster` on different systems and make sure that it comes to your postmaster's mailbox.**

▼ How to Check Aliases

To verify aliases and whether mail can be delivered to a given recipient:

♦ **Type `/usr/lib/sendmail -v -bv` *recipient* and press Return.**

The command displays the aliases and identifies the final address as deliverable or not.

Here is an example of the output:

```
% /usr/lib/sendmail -v -bv sandy
sandy... aliased to   ssmith
ssmith... aliased to              sandy@phoenix
sandy@phoenix... deliverable: mailer esmtp, host phoenix, user sandy@phoenix.eng.acme.com
%
```

You should take extra care to avoid loops and inconsistent databases when both local and domain-wide aliases are used. Be especially careful when you move a user from one system to another to avoid creating alias loops.

▼ How to Test the `sendmail` Rule Sets

1. **Type `/usr/lib/sendmail -bt` and press Return.**

   Information is displayed.

2. **At the last prompt (>) type a `3,0` (zero) and the mail address you want to test.**

3. **Type Control-d to end the session.**

Here is an example of the output:

```
% /usr/lib/sendmail -bt
ADDRESS TEST MODE (ruleset 3 NOT automatically invoked)
Enter <ruleset> <address>
> 3,0 sandy@phoenix
rewrite: ruleset   3   input: sandy @ phoenix
rewrite: ruleset  96   input: sandy < @ phoenix>
rewrite: ruleset  96 returns: sandy < @ phoenix . eng . acme . com . >
rewrite: ruleset   3 returns: sandy < @ phoenix . eng . acme . com . >
rewrite: ruleset   0   input: sandy < @ phoenix . eng . acme . com . >
rewrite: ruleset 199   input: sandy < @ phoenix . eng . acme . com . >
rewrite: ruleset 199 returns: sandy < @ phoenix . eng . acme . com . >
rewrite: ruleset  98   input: sandy < @ phoenix . eng . acme . com . >
rewrite: ruleset  98 returns: sandy < @ phoenix . eng . acme . com . >
rewrite: ruleset 198   input: sandy < @ phoenix . eng . acme . com . >
rewrite: ruleset 198 returns: $# local $: sandy
rewrite: ruleset   0 returns: $# local $: sandy
```

See Chapter 3, for a complete description of the diagnostic information.

▼ How to Verify Connections to Other Systems

To verify connections to other systems, you can use the mconnect program to open connections to other sendmail systems over the network. The mconnect program runs interactively. You can issue various diagnostic commands. See the **mconnect**(1) man page for a complete description. The following example verifies that mail to the user name shamira is deliverable.

```
$ mconnect phoenix
connecting to host phoenix (129.144.52.96), port 25
connection open
220 phoenix.Eng.Acme.COM Sendmail 8.9.0+Sun/8.9.0; Tue, 25 Jul 1998 10:45:28 -0700
vrfy sandy
250 Sandy Smith <sandy@phoenix.Eng.Acme.COM>
>
```

If you cannot use mconnect to connect to an SMTP port, check these conditions:

- Is the system load too high?
- Is the sendmail daemon running?
- Does the system have the appropriate /etc/mail/sendmail.cf file?
- Is port 25 (the port that sendmail uses) active?

System Log

The mail services log most errors using the syslogd program. The default is for syslogd to send messages to the loghost.

You can define a system called loghost in the /etc/hosts file to hold all logs for an entire NIS domain. The system log is supported by the syslogd program. You specify a loghost in /etc/hosts. If no loghost is specified, then error messages from syslogd are not reported.

Code Example 2–1 shows the default /etc/syslog.conf file:

CODE EXAMPLE 2–1 Default /etc/syslog.conf File

```
#ident "@(#)syslog.conf   1.3        93/12/09 SMI"  /* SunOS 5.0 */  #
# Copyright (c) 1994 by Sun Microsystems, Inc.
#
# syslog configuration file.
#
# This file is processed by m4 so be careful to quote (‘’) names
# that match m4 reserved words. Also, within ifdef’s, arguments
# containing commas must be quoted.
#
# Note: Have to exclude user from most lines so that user.alert
#       and user.emerg are not included, because old sendmails
#       have no 4.2BSD based systems doing network logging, you
#       can remove all the special cases for "user" logging.
# *.err;kern.debug;auth.notice;user.none         /dev/console
*.err;kern.debug;daemon,auth.notice;mail.crit;user.none /var/adm/messages
*.alert;kern.err;daemon.err;user.none            operator
*.alert;user.none                                root
*.emerg;user.none                                *
# if a non-loghost machine chooses to have authentication messages
# sent to the loghost machine, un-comment out the following line:
#auth.notice         ifdef(‘LOGHOST’, /var/log/authlog, @loghost)
mail.debug           ifdef(‘LOGHOST’, /var/log/syslog, @loghost)
#
# non-loghost machines will use the following lines to cause "user"
# log messages to be logged locally.
#
ifdef(‘LOGHOST’, ,
user.err             /dev/console
user.err             /var/adm/messages
user.alert           ‘root, operator’
user.emerg           *
)
```

You can change the default configuration by editing the /etc/syslog.conf file. You must restart the syslog daemon for any changes to take effect. You can add these selections to the file to gather information about mail:

- mail.alert—Messages about conditions that should be fixed now
- mail.crit—Critical messages

- mail.warning—Warning messages
- mail.notice—Messages that are not errors, but might need attention
- mail.info—Informational messages
- mail.debug—Debugging messages

The following entry sends a copy of all critical, informational and debug messages to /var/log/syslog.

```
mail.crit;mail.info;mail.debug   /var/log/syslog
```

Each line in the system log contains a time stamp, the name of the system that generated it, and a message. The syslog file can log a large amount of information.

The log is arranged as a succession of levels. At the lowest level, only unusual occurrences are logged. At the highest level, even the most mundane and uninteresting events are recorded. As a convention, log levels under 10 are considered "useful." Log levels higher than 10 are usually used for debugging. See the "Customizing System Message Logging" in *System Administration Guide, Volume II* for information about loghost and the syslogd program.

Other Diagnostic Information

For other diagnostic information, check the following sources:

- Look at the Received lines in the header of the message. These lines trace the route the message took as it was relayed. Notice that in the UUCP network many sites do not update these lines, and in the Internet the lines often get rearranged. To straighten them out, look at the date and time in each line. Remember to account for time–zone differences.
- Look at the messages from MAILER-DAEMON. These typically report delivery problems.
- Check the system log that records delivery problems for your group of systems. The sendmail program always records what it is doing in the system log. You might want to modify the crontab file to run a shell script nightly that searches the log for SYSERR messages and mails any that it finds to the postmaster.
- Use the mailstats program to test mail types and determine the number of incoming and outgoing messages.

CHAPTER 3

sendmail Reference

The sendmail program is a mail transport agent that uses a configuration file to provide aliasing and forwarding, automatic routing to network gateways, and flexible configuration. The Solaris operating environment supplies standard configuration files that most sites can use. Chapter 2 explains how to set up an electronic mail system using the standard files. This chapter describes some of the differences between the generic version of sendmail and the Solaris version.

What's New with sendmail

Version 8.9 of sendmail has been included with the Solaris 7 release. Here is a list of the important or user-visible changes that are included in this new version:

- A new system for building configuration files. Instructions for using the new system is included in "Building a sendmail Configuration File" on page 32.
- The permissions and the ownership of several directories have been changed in order to increase security. When the Solaris 7 release is installed, /etc/mail and /var/spool/mqueue and the parent directories will have the correct permissions.
- Increased security on .forward files requires that the default shells (as listed in /etc/passwd) of all users trying to employ a .forward file to forward mail to a

program or to a file must be listed in /etc/shells for the file to be accessed. See "How to Create and Populate /etc/shells" on page 44 for more information.

- Additional restrictions have been put in place on .forward and :include: files. These files and the directory structure that they are placed in cannot be group- or world-writable. A script called /usr/lib/mail/sh/check-permissions is included to help identify files with unsafe permissions.
- The use of .forward files has been enhanced. A .forward.*hostname* file can be used to reroute mail sent to a user at a specific host. Also, a .forward+*detail* file can be used to determine who is using an alias. These files are described in ".forward Files" on page 66.
- The way sendmail acts when an owner alias exists has changed. A full description of the change can be found in "Mailbox" on page 6. You can download a script called check-aliases.sh, which checks all alias files listed in /etc/mail/sendmail.cf for misconfigured owner-aliases.
- The sendmail program requires a fully qualified host name when starting. A script called /usr/lib/mail/sh/check-hostname is included with the release to identify host configurations that do not support fully qualified host names.

Additional information on the Solaris version of sendmail can be found at http://www.sendmail.org/sun-specific/migration+sun.html.

The New Configuration Files

In order to customize your mail system, it can be necessary to re-configure sendmail. Earlier Solaris releases contained a large file that included many cryptic options, that needed to be manually edited to make any changes to the way sendmail functions. In the Solaris 7 release, a new configuration system has been included, which uses m4 to build the configuration file (see the **m4**(1) man page).

Changes to the sendmail Command Line Options

The options listed in the following table are the new options for the Solaris 7 release. A complete description of these options can be found in *sendmail, Second Edition*, by Bryan Costales.

TABLE 3–1 sendmail Command-Line Arguments Changes

Argument	Description
–bD	Run as a daemon, but do not fork so that sendmail always runs in the foreground.
–bH	Purge persistent host status.
–bh	Print persistent host status.
–M	Assign a macro value.
–N	Append the DSN NOTIFY command to the ESMTP RCPT command.
–O	Use to set a multicharacter configuration option.
–p	Set the protocol and hostname.
–R	Include the DSN RET command to the ESMTP MAIL command.
–U	Used to indicate that this is the very first step in this submission.
–V	Specify the envelope indentifier for outgoing messages.

Changes to the sendmail Configuration File Options

The options listed in the following table are the new configuration options for the Solaris 7 release. These options are sorted by their multicharacter name. If the option still has a single character name, it is displayed parenthetically. Many of the single character options supported in 2.6 are still supported in the Solaris 7 release. A complete description of these options can be found in *sendmail, Second Edition*, by Bryan Costales.

TABLE 3–2 `sendmail` Configuration File Option Changes

Argument	Description
`AllowBogusHELO`	Allow no hostname with `HELO` or `EHLO`.
`ColonOkInAddr`	Allow colons in addresses.
`ConnectionRateThrottle`	Slow the acceptance rate of new connections.
`DefaultCharSet`	Define default character set.
`DialDelay`	Set delay time for second `connect()` attempt.
`DontBlameSendmail`	Disable parts of security checking.
`DontExpandCnames`	Prevent canonical name expansion.
`DontInitGroups`	Do not use `initgroups()`.
`DontProbeInterfaces`	Disable automatic probing of interfaces.
`DoubleBounceAddress`	Set email address for error notifications.
`EightBitMode`	Specify how to handle unlabeled 8–bit data.
`ErrorHeader` (E)	Append custom text ahead of error message text.
`ForwardPath` (J)	Set alternative locations of the `.forward` file.
`HostsFile`	Specify an alternative location for the `/etc/hosts` file.
`HostStatusDirectory`	Set the location of the directory containing persistent host status data.
`MaxDaemonChildren`	Limit the number of forked children of `sendmail`.
`MaxMessageSize`	Set the maximum messages size.
`MaxMimeHeaderLength`	Set the maximum length of certain MIME header field values.
`MaxRecipientsPerMessage`	Set the maximum number of message recipients.

TABLE 3–2 sendmail Configuration File Option Changes *(continued)*

Argument	Description
MaxQueueRunSize	Set the number of queued messages that can be processed in one run.
MinQueueAge	Determine the minimum amount of time a message must be in the queue before processing.
MustQuoteChars	Set the list of characters that must be quoted in nonaddress information.
NoRecipientAction	Determine how to handle headers without recipients.
OperatorChars or $o	Establishe the list of separation operators.
QueueSortOrder	Specify how to sort the queue.
RunAsUser	Run sendmail as a non-root user.
SafeFileEnvironment	Select the directory for safe file writes.
ServiceSwitchFile	Specify the location of the switch file for name services.
SingleLineFromHeader	Convert all newlines in the From: header to space characters.
SingleThreadDelivery	Select single threaded delivery.
UnsafeGroupWrites	Check for unsafe group permissions.

Solaris sendmail Differences

This section describes some of the changes included in the Solaris version of sendmail as compared to the generic Berkeley version.

Flags Used When Compiling sendmail

The following table lists the flags used when compiling the version of sendmail delivered with the Solaris 7 release. If your configuration requires other flags, then you need to download the source and recompile the binary yourself. Information about this process can be found at http://www.sendmail.org.

- SOLARIS=20700 — Support for the Solaris 7 operating environment
- NDBM — Support for ndbm databases
- NEWDB — Support for db databases
- NIS — Support for nis databases
- NISPLUS — Support for nisplus databases
- USERDB — Support for the User database
- MAP_REGEX — Support for regular expression maps
- SUN_EXTENSIONS — Solaris-specific flag; support for Sun-specific extensions included in sun_compat.o
- VENDOR_DEFAULT=VENDOR_SUN — Solaris-specific flag; selects Sun as the default vendor
- USE_VENDOR_CF_PATH — Solaris-specific flag; allows for the configuration file to be placed in /etc/mail
- _FFR_MAXALIASRECURSION_OPTION — Solaris-specific flag; enables selection of MaxAliasRecursion option
- _FFR_MAX_MIME_HEADER_LENGTH — Solaris-specific flag; enables selection of MaxMimeHeaderLength option

Alternative sendmail Commands

The Solaris release does not include all of the command synonyms that are provided in the generic release from Berkley. This table includes a complete list of the command aliases, whether they are included in the Solaris release, and how to generate the same behavior using sendmail.

TABLE 3–3 Alternate sendmail Commands

Alternate Name	Included in Solaris?	Options with sendmail
hoststat	no	sendmail –bh
mailq	yes	sendmail –bp
newaliases	yes	sendmail –bi

TABLE 3–3 Alternate sendmail Commands *(continued)*

Alternate Name	Included in Solaris?	Options with sendmail
purgestat	no	sendmail –bH
smtpd	no	sendmail –bd

Define Configuration File Version

The new version of sendmail (version 8) includes a new configuration option which defines the version of the sendmail.cf file. This will allow older configuration files to be used with Version 8 sendmail. You can set the version level to values between 0 and 8. You can also define the vendor. Either Berkeley or Sun are valid vendor options. If the V option is not defined in the configuration file, the default setting is V1/Sun. If a version level is specified but no vendor is defined, then Sun is used as the default vendor setting. Table 3–4 lists some of the valid options.

TABLE 3–4 Configuration File Version Values

Field	Description
V1/Sun	Use Solaris extensions of name service support. This option allows for old configuration files to be used with the new version of sendmail. This is the default setting if nothing is specified.
V7/Sun	Use for Version 8.8 of sendmail.
V8/Sun	Use for Version 8.9 of sendmail. This is the setting that is included in prebuilt configuration file in the Solaris 7 release.

How Mail Addressing Works

The path a mail message follows during delivery depends on the setup of the client system and the topology of the mail domain. Each additional level of mail hosts or mail domains can add one more round of alias resolution, but the routing process is basically the same on most hosts.

You can set up a client system to receive mail locally or select a remote to receive the mail for the client system. Receiving mail locally is known as running `sendmail` in local mode. Local is the default mode for all mail servers and some clients. If the client is mounting `/var/mail` from a server, then the client is running `sendmail` in remote mode.

Assuming that you are using the default rule set in the `sendmail.cf` file, the following examples show the route an email message takes.

On a mail client in remote mode, a mail message will go through the following routing process:

1. **Expand the mail alias, if possible, and restart the local routing process.**

   The mail address is expanded by looking up the mail alias in the name space, according to entry in `/etc/nsswitch.conf`, and substituting the new value, if one is found. This new alias is then checked again.

2. **If the address cannot be expanded, forward it to the mail server.**

   If the mail address can not be expanded, then there could be a problem with the address or the address is not local. In both cases, the mail server needs to resolve the problem.

3. **If the expanded alias loops back to the original address, forward the mail to the mail server.**

   The process keeps a history of all of the lookups and if the original alias is generated again, the mail is forwarded to the mail server to resolve.

On the mail server or a mail client in local mode, a mail message goes through the following routing process:

1. **Expand the mail alias, if possible, and restart the local routing process.**

   The mail address is expanded by looking up the mail alias in the name space and substituting the new value, if one is found. This new alias is then checked again.

2. **If the mail is local, deliver it to `/usr/lib/mail.local`.**

   The mail will be delivered to a local mailbox.

3. **If the mail address includes a host in this mail domain, deliver the mail to that host.**

4. **If the address does not include a host in this domain, forward the mail to the mail host.**

   The mail host uses the same routing process as the mail server, but the mail host can receive mail addressed to the domain name as well as to the host name.

How sendmail Interacts With a Name Service

Mail domain is a concept used by the standard sendmail.cf file to determine whether mail should be delivered directly or through the mail host. Intra-domain mail is delivered through direct SMTP connection, while inter-domain mail is forwarded to a mail host.

In a secure network, only a few selected hosts are authorized to generate packets targeted to external destinations. Even if a host has the IP address of the remote host external to the mail domain, this does not guarantee that an SMTP connection can be established. The standard sendmail.cf assumes the following:

- The current host is not authorized to send packets directly to a host outside the mail domain.
- The mail host is capable of forwarding the mail to an authorized host that can transmit packets directly to an external host. (In fact, the mail host can itself be an authorized host.)

Given these assumptions, it is the responsibility of the mail host to deliver or forward inter-domain mail.

Setting Up sendmail Requirements for Name Services

sendmail imposes various requirements on name services. This section explains these requirements and how to satisfy them. For more information, refer to the **in.named**(1M), **NIS+**(1), **nisaddent**(1M), and **nsswitch.conf**(4) man pages.

Establishing the Mail Domain Name with a Name Service

The mail domain name must be a suffix of the name service domain. For example, if the domain name of the name service is A.B.C.D, then the mail domain name could be one of the following:

- A.B.C.D
- B.C.D
- C.D
- D

When first established, the mail domain name is often identical to the name service domain. As the network grows larger, the name service domain can be divided into smaller pieces to make the name service more manageable. However, the mail domain often remains undivided to provide consistent aliasing.

Host Name Space Data

The host table or map in the name service must be set up to support three types of `gethostbyname()` queries:

- `mailhost` – Some name service configurations satisfy this requirement automatically.
- Full host name (for example, `smith.admin.acme.com`) – Many name service configurations satisfy this requirement.
- Short host name (for example, `smith`) – `sendmail` must connect to the mail host to forward external mail. To determine if a mail address is within the current mail domain, `gethostbyname()` is invoked with the full host name. If the entry is found, the address is considered internal.

  NIS, NIS+, and DNS all support `gethostbyname()` with a short host name as an argument, so this requirement is automatically satisfied.

Two additional rules about the host name space need to be followed to properly establish the sendmail services within a name space.

1. `gethostbyname()` with full and short host name should yield consistent results. For example, `gethostbyname(smith.admin.acme.com)` should return the same result as `gethostbyname(smith)` as long as both functions are called from the mail domain `admin.acme.com.`
2. For all name service domains under a common mail domain, `gethostbyname()` with a short host name should yield the same result. For example, given the mail domain `smith.admin.acme.com`, `gethostbyname(smith)` should return the same result calling from either domain `ebb.admin.acme.com` or `esg.admin.acme.com`. The mail domain name is usually shorter than the name service domain, giving this requirement special implications for various name services.

Configuration Issues with NIS and `sendmail`

This list includes all the configuration issues that you must resolve before using `sendmail`, when using NIS as your only name service.

mail domain name — If you are setting up NIS as the primary name service, `sendmail` automatically strips off the first component of the NIS domain name and uses the result as mail domain name. For example, `ebs.admin.acme.com` becomes `admin.acme.com`.

mail host host name — You must have a `mailhost` entry in the NIS host map.

full host names — The normal NIS setup does not “understand” the full host name. Rather than trying to make NIS understand the full host name, turn off this requirement from the sendmail side by editing the sendmail.cf file and replacing all occurrences of %l with %y. This turns off sendmail's inter-domain mail detection. As long as the target host can be resolved to an IP address, a direct SMTP delivery is attempted. Make sure that your NIS host map does not contain any host entry that is external to the current mail domain. Otherwise, you need to further customize the sendmail.cf file.

matching full and short host names — Follow the previous instructions on how to turn off gethostbyname() for a full host name.

multiple NIS domains in one mail domain — All NIS host maps under a common mail domain should have the same set of host entries. For example, the host map in the ebs.admin.acme.com domain should be the same as the host map in the esg.admin.acme.com. Otherwise, one address might work in one NIS domain but fail in the other NIS domain.

Configuration Issues with NIS and DNS while Using sendmail

This list includes all the configuration issues that you must resolve before using sendmail, when using NIS with DNS as your name service.

mail domain name — If you are setting up NIS as the primary name service, sendmail automatically strips the first component of the NIS domain name and uses the result as mail domain name, for example, ebs.admin.acme.com becomes admin.acme.com.

mailhost host name — When the DNS forwarding feature is turned on, queries that NIS cannot resolve are forwarded to DNS, so there is no need for a mailhost entry in the NIS host map.

full host names — Although NIS does not "understand" full host names, DNS does. This requirement is satisfied when you follow the regular procedure for setting up NIS and DNS.

matching full and short host names — For every host entry in the NIS host table, you must have a corresponding host entry in DNS.

multiple NIS domains in one mail domain — All NIS host maps under a common mail domain should have the same set of host entries. For example, the host map in the ebs.admin.acme.com domain should be the same as the host map in the esg.admin.acme.com. Otherwise, one address might work in one NIS domain but fail in the other NIS domain.

Configuration Issues with NIS+ and sendmail

This list includes all the configuration issues that you must resolve before using sendmail when using NIS+ as your only name service.

mail domain name — If you are setting up NIS+ as your primary name service, sendmail can look up the mail domain from the NIS+ sendmailvars table, a two–column NIS+ table with one key column and one value column. To set up your mail domain, you must add one entry to this table. This entry should have the key column set to the literal string maildomain and the value column set to the your mail domain name (for example, admin.acme.com). Although NIS+ allows any string in the sendmailvars table, the suffix rule still applies for the mail system to work correctly. You can use nistbladm to add the maildomail entry to the sendmailvars table. For example:

```
nistbladm -A key="maildomain" value=<mail domain> sendmailvars.org_dir.<NIS+ domain>
```

Notice that this mail domain is a suffix of the NIS+ domain.

mailhost host name — You must have a mailhost entry in the NIS+ hosts table.

full host names — NIS+ “understands” the full host name. Following the regular NIS+ setup procedure satisfies this requirement.

matching full and short host names — To satisfy this requirement, you can duplicate the entries in the host table, or you can enter all host entries in the user name service domains into a master host table at mail domain level.

multiple NIS domains in one mail domain — To satisfy this requirement, you can duplicate the entries in all the host tables, or you can type all host entries in the user name service domains into a master host table at mail domain level. Because you are merging (logical or physical) multiple host tables into one host table, the same host name cannot be reused in the multiple name service domain sharing a common mail domain.

Configuration Issues with NIS+ and DNS while Using sendmail

This list includes all the configuration issues that you must resolve before using sendmail when using NIS+ with DNS as your name service.

mail domain name — If you are setting up NIS+ as your primary name service, sendmail can look up the mail domain from the NIS+ sendmailvars table, a two–column NIS+ table with one key column and one value column. To set up your mail domain, you must add one entry to this table. This entry should have the key column set to the literal string maildomain and the value column set to the your mail domain name (for example, admin.acme.com). Although NIS+ allows any string in the sendmailvars table, the suffix rule still applies for the mail system to work correctly. You can use nistbladm to add the maildomail entry to the sendmailvars table. For example:

```
nistbladm -A key="maildomain" value=<mail domain> sendmailvars.org_dir.<NIS+ domain>
```

Notice that this mail domain is a suffix of the NIS+ domain.

mailhost host name — If your network uses both NIS+ and DNS as the source for the host database, you can put the `mailhost` entry in either the NIS+ or DNS host table. Make sure that your users list NIS+ and DNS as the source for the host database in the `/etc/nsswitch.conf` file.

full host names — Both NIS+ and DNS “understand” full host names. Following the regular NIS+ and DNS setup procedures satisfies this requirement.

matching full and short host names — For every host entry in the NIS+ host table, you must have a corresponding host entry in DNS.

multiple NIS domains in one mail domain — To satisfy this requirement, you can duplicate the entries in all the host tables, or you can type all host entries in the user name service domains into a master host table at the mail domain level.

Other Mail-Related Topics

This section includes specific information about the various forms of alias files that can be used on a Solaris system. In addition, a discussion of `.forward` files is included.

Mail Alias Files

You can use any of the following files to maintain aliases. Which type of file to use depends on who will be using the alias and who needs to be able to change the alias. Each type of alias file has unique format requirements. Each of these is defined in the following sections.

`.mailrc` Aliases

Aliases listed in a `.mailrc` file are accessible only by the user who owns the file. This allows users to establish an alias file they control and that is usable only by its owner. Aliases in a `.mailrc` file adhere to the following format:

```
alias aliasname value value value ...
```

where *aliasname* is the name the user will use when sending mail, and *value* is a valid email address.

If a user establishes a personal alias for `scott` that does not match the email address for `scott` in the name space, mail will be routed to the wrong person when other people try to reply to mail generated by that user. The only workaround is to use any of the other aliasing mechanisms.

/etc/mail/aliases

Any alias established in the `/etc/mail/aliases` file can be used by any user who knows the name of the alias and the host name of the system that contains the file. Distribution list formats in a local `/etc/mail/aliases` file adhere to the following format:

```
aliasname: value,value,value...
```

where *aliasname* is the name the user will use when sending mail to this alias and *value* is a valid email address.

The aliases in the `/etc/mail/aliases` file are stored in text form. When you edit the `/etc/mail/aliases` file, run the `newaliases` program to recompile the database and make the aliases available in binary form to the `sendmail` program. Or you can use Administration Tool's Database Manager to administer the mail aliases stored in local `/etc` files.

Normally, the root user only can edit this file. If using the Administration Tool, then all users in group 14, which is the sysadmin group, will be able to change the local file. Another option is to create an entry like:

```
aliasname: :include:/path/aliasfile
```

where *aliasname* is the name the user will use when sending mail and */path/aliasfile* is the full path to the file that includes the alias list. The alias file should include email entries, one entry on each line, and no other notations:

```
user1@host1
user2@host2
```

You can define additional mail files in `/etc/mail/aliases` to keep a log or a backup copy. The following entry stores all mail sent to *aliasname* in *filename.*

```
aliasname: /home/backup/filename
```

You can also route the mail to another process. The following stores a copy of the mail message in *filename* and prints a copy.

```
aliasname: "|tee -a /home/backup/filename |lp"
```

NIS Aliases Map

All users in the local domain can use entries included in the NIS aliases map. The `sendmail` program can use the NIS aliases map instead of the local `/etc/mail/aliases` files to determine mailing addresses. See the **`nsswitch.conf`**(4) man page for more information.

Aliases in the NIS `aliases` map adhere to the following format:

```
aliasname: value,value,value...
```

where *aliasname* is the name the user will use when sending mail and *value* is a valid email address.

The NIS aliases map should contain entries for all mail clients. In general, only the root user on the NIS master can change these entries. This type of alias might not be a good choice for aliases that are constantly changing, but can be useful if the alias points to another alias file; as in this syntax example:

```
aliasname: aliasname@host
```

where *aliasname* is the name the user will use when sending mail and *host* is the host name for the server that contains an `/etc/mail/alias` file.

NIS+ `mail_aliases` Table

The NIS+ `mail_aliases` table contains the names by which a system or person is known in the local domain. The `sendmail` program can use the NIS+ mail_aliases table instead of the local `/etc/mail/aliases` files to determine mailing addresses. See the **`aliasadm`**(1M) and **`nsswitch.conf`**(4) man pages for more information.

Aliases in the NIS+ mail_aliases table adhere to the following format:

```
alias:    expansion     [options   # "comments"]
```

Table 3–5 describes the four columns.

TABLE 3–5 Columns in the NIS+ mail_aliases Table

Column	Description
alias	The name of the alias
expansion	The value of the alias or a list of aliases as it would appear in a sendmail /etc/mail/aliases file
options	Reserved for future use
comments	Comments about an individual alias

The NIS+ mail_aliases table should contain entries for all mail clients. You can list, create, modify, and delete entries in the NIS+ aliases table with the aliasadm command. Or you can use Administration Tool's Database Manager to administer NIS+ mail aliases.

If you are creating a new NIS+ aliases table, you must initialize the table before you create the entries. If the table exists, no initialization is needed. See "To List Individual Entries in the NIS+ mail_aliases Table " on page 34 for information about how to create a NIS+ mail_aliases table.

To use the aliasadm command, you must be a member of the NIS+ group that owns the aliases table or the person who created the table.

.forward Files

Users can create a .forward file in their home directories that sendmail uses to temporarily redirect mail or send mail to a custom set of programs without consulting a system administrator. When troubleshooting mail problems, particularly problems with mail not being delivered to the expected address, always check the user's home directory for a .forward file.

A common mistake users make is to put a .forward file in the home directory of host1 that forwards mail to user@host2. When the mail gets to host2, sendmail looks up user in the NIS or NIS+ aliases and sends the message back to user@host1, resulting in a loop, and more bounced mail.

Note - The root and bin accounts should never have .forward files. Creating these files will create a large security hole. If necessary, forward mail using the aliases file instead.

In order for a .forward file to be consulted during the delivery of mail, the file must be writable only by the owner of the file. This prevents other users from breaking security. In addition, the paths leading up to the home directory must be owned and writable by `root` only. In particular, if a `.forward` file is in `/export/home/terry`, then `/export` and `/export/home` must be owned and writable only by `root`. The actual home directory should be writable only by the user. Other restrictions on a `.forward` file are that the file cannot be a symbolic link and cannot have more than one hard link.

In addition to the standard `.forward` file, a `.forward.`*hostname* file can be created to redirect mail sent to a specific host. For example, if a user's alias has changed from a `sandy@phoenix.eng.acme.com` to `sandy@eng.acme.com`, then it might be nice to place a `.forward.phoenix` file in the home directory for `sandy`.

```
% cat .forward.phoenix
sandy@eng.acme.com
"|/usr/bin/vacation sandy"
% cat .vacation.msg
From: sandy@eng.acme.com (via the vacation program)
Subject: my alias has changed

My alias has changed to sandy@eng.acme.com.
Please use this alias in the future.
The mail that I just received from you
has been forwarded to my new address.

Sandy
```

This allows for the mail to be forwarded to the correct place while also notifying the sender of the alias change. Notice that since the `vacation` program allows only one message file, so this can be done for only one message at a time. However, if the message is not host-specific, one vacation message file can be used by `.forward` files for many hosts.

Another extension to the forwarding mechanism is the `.forward+`*detail* file. The *detail* string can be any sequence of characters as long as no operator characters are used. The operator characters are `.:%&!^[]+`. Using a file like this can make it possible to determine if someone else is giving your email address away. For instance, if a user told someone to use the email address `sandy+test1@eng.acme.com`, the user would be able to identify any future mail that was delivered to this alias. By default, any mail sent to `sandy+test1@eng.acme.com` alias is checked against the alias and `.forward+`*detail* files. If there are no matches, then the mail falls back to delivery to `sandy@eng.acme.com`, but the user is able to see a change in the `To:` header in their mail.

Index