HP-UX Mailing Services Administrator's Guide

HP-UX 11i v1 and HP-UX 11i v2

Manufacturing Part Number: 5991-6611

July 2006

Legal Notices

The information in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Warranty

A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.

U.S. Government License

Proprietary computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

Trademark Notices

UNIX is a registered trademark of The Open Group.

Intel and Itanium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Contents

Contents

Contents

About This Document

This manual describes the Mailing Services implemented in the HP-UX 11i v2 operating system.It is one of the five manuals documenting the Internet Services suite of products. See “Related Documentation” on page 13 for a list of the other new Internet Services manuals. These manuals replace the manual *Installing and Administering Internet Services* (B2355-90685), which was shipped with previous releases of the operating system.

This manual assumes that the HP-UX 11i v2 operating system and the appropriate files, scripts, and subsets are installed.

New Documentation Changes in this Edition

This manual is updated to include the time zone information in `mailx`(1). For more information, see “The mailx Utility” on page 22.

Intended Audience

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

HP-UX Release Name and Release Identifier

Each HP-UX 11i release has an associated release name and release identifier. The `uname`(1) command with the `-r` option returns the release identifier. Table 1 shows the releases available for HP-UX 11i.

Table 1 **HP-UX 11i Releases**

Release Identifier	Release Name	Supported Processor Architecture
B.11.11	HP-UX 11i v1	PA-RISC

Table 1 **HP-UX 11i Releases (Continued)**

Release Identifier	Release Name	Supported Processor Architecture
B.11.20	HP-UX 11i v1.5	Intel® Itanium® Processor Family
B.11.22	HP-UX 11i v1.6	Intel® Itanium® Processor Family
B.11.23	HP-UX 11i v2.0	Intel® Itanium® Processor Family PA-RISC

Publishing History

Table 2 provides, for a particular document, the manufacturing part number, the respective operating systems, and the publication date.

Table 2 **Publishing History Details**

Document Manufacturing Part Number	Operating System Supported	Publication Date
B2355-90110	10.x	June 1996
B2355-90147	11.0	October 1997
B2355-90685	11i v1 11i v1.5 11i v1.6	December 2000
B5969-4360	11i v1.6	April 2002
B2355-90776	11i v2	September 2004
5991-0707	11i v1 11i v2	February 2005

What Is In This Document

HP-UX Mailing Services Administrator's Guide is divided into chapters, which contain information about different mailing services and the Sendmail configuration.

Table 3 describes the content in more detail.

Table 3 Document Organization

Chapter	Description
Mailing Services Overview	Provides an overview of the Mail User Agents and the Mail Transport Agent implemented in the HP-UX 11i v2 operating system.
Configuring and Administering Sendmail	Describes the various steps involved in configuring Sendmail. This section also provides a brief description of how Sendmail works, the Sendmail configuration file, Sendmail restricted shell (`smrsh`), and some troubleshooting measures for Sendmail.
Sendmail 8.13.3	Describes the new features in Sendmail 8.13.3, which is released as a Web upgrade on the HP-UX 11i v1 and HP-UX 11i v2 operating systems.

Related Documentation

For more information about the Internet Services suite of products, see the following books:

- *Sendmail 8.13.3 Release Notes*

  Provides a brief description of the new and changed features in Sendmail 8.13. You can access this document at the following URL:

  http://www.docs.hp.com/hpux/netcom/index.html#Internet%20Services

- *Sendmail 8.13.3 Programmer's Guide*

  Provides a detailed description of the Mail Filter (MILTER) APIs, which is a new feature in Sendmail 8.13.3. You can access this document at the following URL:

  http://www.docs.hp.com/hpux/netcom/index.html#Internet%20Services

- *HP-UX Internet Services Administrator's Guide*

  Provides an overview of the Internet Services products and describes how to install and configure them on your HP-UX 11i v2 operating system. You can access this manual at the following URL:

  http://www.docs.hp.com/hpux/netcom/index.html#Internet%20Services

- *HP-UX Routing Services Administrator's Guide*

  Provides an overview of the routing daemons, `gated` and `mrouted`, supported in the HP-UX 11i v2 operating system. It also explains the various protocols that these routing daemons support.You can access this manual at the following URL:

  http://www.docs.hp.com/hpux/netcom/index.html#Internet%20Services

- *HP-UX IP Address and Client Management Administrator's Guide*

  Provides an overview of the IP address and client management implementations on the HP-UX 11i v2 operating system, where BIND, DHCPv6 and SLP deals with the client management, and NTP deals with the IP address management. You can access this manual at the following URL:

  http://www.docs.hp.com/hpux/netcom/index.html#Internet%20Services

- *HP-UX Remote Access Services Administrator's Guide*

  Provides information about the Remote Access Services available in the HP-UX 11i v2 operating system: r-commands, WU-FTP, and `telnet`. You can access this manual at the following URL:

  http://www.docs.hp.com/hpux/netcom/index.html#Internet%20Services

- Request for Comments (RFC)

  Many sections of this manual refer to RFCs for more information about certain networking topics. These documents publicize Internet standards, new research concepts, and status memos about the Internet. You can access the full range of RFC documents and more information about the Internet Engineering Task Force (IETF) at the following URL:

  http://www.ietf.org/rfc.html

- Other Documents

  For detailed technical and conceptual information about BIND, as well as information about planning a BIND hierarchy and using Sendmail with BIND, HP recommends that you read *DNS and BIND*, by Paul Albitz and Cricket Liu, published by O'Reilly and Associates, Inc. You can get information about the book (including retail outlets where you can buy it, as well as how to order it directly from O'Reilly) by visiting the O'Reilly Website

  `http://www.ora.com`

Typographical Conventions

This document uses the following typographic conventions:

`audit` (5)	An HP-UX manpage. In this example, *audit* is the name and *5* is the section in the *HP-UX Reference*. On the web and on the Instant Information CD, it may be a hot link to the manpage itself. From the HP-UX command line, you can enter "`man audit`" or "`man 5 audit`" to view the manpage. See `man` (1).
Book Title	The title of a book. On the web and on the Instant Information CD, it may be a hot link to the book itself.
`ComputerOut`	Text displayed by the computer.
`Command`	A command name, qualified command phrase, daemon, file, or option name.
$	The system prompt for the Bourne, Korn, and POSIX shells.
#	The superuser prompt.
`Variable`	The name of a variable that you may replace in a command or function or information in a display that represents several possible values.
[ ] { }	In syntax definitions, square brackets indicate items that are optional and braces indicate items that are required.
(`Ctrl+A`)	This symbol indicates that you hold down the first named key while pressing the key or mouse button that follows the plus.

HP Encourages Your Feedback

HP welcomes any comments and suggestions you have on this manual.

You can send your comments in the following ways:

- Internet electronic mail: netinfo_feedback@cup.hp.com
- Using a feedback form located at the following URL:

  http://docs.hp.com/assistance/feedback.html

Please include the following information along with your comments:

- The full title of the manual and the part number. (The part number appears on the title page of printed and PDF versions of a manual.)
- The section numbers and page numbers of the information on which you are commenting.
- The version of HP-UX that you are using.

1 Mailing Services Overview

Mailers are a set of UNIX commands that provide command-line interfaces for users to send and receive messages over the network. These interfaces, which are generally referred to as **Mail User Agents**

(MUA), communicate with a **Mail Transport Agent** (MTA) to send mail messages to the appropriate destination, and receive messages destined to the end user's mailbox.

An MUA is a program that allows users to compose and read electronic mail messages. The MUA provides an interface between the user and the MTA. An outgoing mail is eventually delivered to an MTA for delivery, and the incoming messages are collected from the MTA.

An MTA is a program that is responsible for delivering electronic mail messages. Upon receiving a message from an MUA or another MTA, an MTA stores the message locally, analyzes the recipients, and either delivers the message (for local addresses) or forwards the message to another MTA for routing. In either case, the MTA can edit and add to the message headers.

HP-UX systems use the Sendmail MTA and the `elm`, `mail`, and `mailx` MUAs.

Table 1-1 lists the MTA and MUAs that HP-UX 11i v2 supports.

Table 1-1 MTA and MUAs Supported on HP-UX 11i v2

MTA/MUA	Description
`elm`	`elm` is the electronic mail processing system for UNIX. It is designed as an MUA to run with Sendmail to send or receive messages. The most significant difference between `elm` and other mail systems is that it is screen-oriented.
`mail`/`rmail`	`mail`/`rmail` is a customized HP program used to send remote or local mail. It is primarily used by Sendmail for local mail delivery.
`mailx`	`mailx` is an interactive message processing system that provides a comfortable and flexible environment for sending and receiving messages electronically.
`Sendmail`	`Sendmail` sends a message to one or more recipients or addresses, routing the message over appropriate networks.

This chapter discusses the following topics:

- `elm`
- `mailx`
- `mail/rmail`
- `sendmail`

CAUTION Do not use two separate mail programs simultaneously to access the same mail file. This may cause unpredictable results.

The elm Utility

The `elm` utility is based on the public domain `elm` program. An electronic mail for UNIX, `elm` is a Mail User Agent (MUA) system designed to run with Sendmail or with any other UNIX MTA configured on your system.

The `elm` program is a screen-oriented mail processing system that includes the following features:

- An industry-wide MIME standard for nontext mails
- A special forms message and form reply mechanism
- An easy-to-use alias system for individuals and groups

`elm` operates in three principal modes:

- Interactive mode – Executes as an interactive mail interface program.
- Message mode – Sends a single interactive message to a list of mail addresses – from the command prompt.
- File mode – Sends a file or command output to a list of mail addresses from the command line or by using redirection.

When `elm` operates in any of these modes, `elm` honors the values set in the `$HOME/.elm/elmrc` initialization file, `elm` alias database, and the system `elm` alias database.

How elm Works

`elm`'s screen-oriented mail processing interface displays all the options necessary to send and compose messages on the screen. You can select the most appropriate option based on your requirement.

When invoked, `elm` first displays the main or message menu. `elm` reads customized variables from the `$HOME/.elm/elmrc` file to initialize the parameters. The main menu displays index entries for the messages in your inbox or selected mail folder. Among other options, you can read, print, reply to, and forward these messages, as well as initiate new mail messages to other users. Some commands use a series of prompts to complete their action. You can use the Ctrl-D keys to cancel their operations.

For a detailed description of all the commands used to edit and send mail messages, type man 1M elm at the HP-UX prompt.

The elm Configuration File

The elm configuration file, $HOME/.elm/elmrc, defines the initial values for the elm configuration variables. You can create the configuration file by choosing the o option (the options menu) in the main menu, which displays a list of all the elm configuration variables. Choose the appropriate option in the options menu to modify the configuration variable.

When invoked, elm reads the customized variables from the $HOME/.elm/elmrc file to initialize the parameters.

The following types of configuration variables are available in the elm configuration file:

- String – String variables have the following form:

  ```
  string-name = string-value
  ```

- Numeric – Numeric variables have the following form:

  ```
  numeric_variable- name = numeric value
  ```

- Boolean – Boolean variables have the following form:

  ```
  boolean-name = ON
  ```

  and

  ```
  boolean-name = OFF
  ```

Some examples of elm variables follow:

```
N>ames only : OFF

U>ser level : Beginning User
```

The $HOME/.elm/elmrc file can contain any combination of the string, numeric, and Boolean variables.

For a detailed description of the numeric, string and boolean variables, type man 1 elm at the HP-UX prompt.

The mailx Utility

`mailx` is an interactive message processing system. It provides a flexible environment for sending and receiving messages electronically. `mailx` provides commands to save, delete, and reply to messages.

You can use `mailx` to edit, review, and modify messages. By default, incoming mail is stored in a standard file called a **system mailbox**, unless you specify an alternate mailbox file using the `-f` option. As incoming messages are read from the system mailbox, they are marked to be moved to a secondary file for storage. When you exit from `mailx`, these marked messages are moved to the secondary storage file. Hence these messages are not displayed the next time `mailx` is invoked. Messages remain in this file until removed explicitly.

During startup, `mailx` reads commands from a system-wide file, `/usr/share/lib/mailx.rc`, to initialize certain parameters. It then uses the personalized variables available in the user-specific startup file, `$HOME/mailrc`. When you invoke `mailx`, a header summary of all the messages is displayed, followed by a prompt indicating that `mailx` can accept regular options. Each message is assigned with a sequential number, and the first message is always marked by a > in the header summary.

`mailx` operates in command mode when you read mail and in input mode when you send mail. The behavior of `mailx` is governed by a set of environment variables, flags, and valued parameters that you can enable and disable using the `set` and `unset` options.

`mailx` provides a list of options, environment variables, and tilde escape commands. You can use tilde escape commands only in input mode by beginning a line with the tilde escape character (~). Environment variables are internal `mailx` program variables, and can be imported from the execution environment.

`mailx` provides native language support (NLS) for processing mails in different languages. To enable NLS support for a language, the respective language definition must exist in the HP-UX system. In an NLS environment, `mailx` depends on the time zone information defined in the mail header to display the date and time information. Table 1-2 lists the time zones currently supported by `mailx`.

Table 1-2 Time Zones Supported by mailx

nst	ast	adt	est	edt	cst	cdt	mst	mdt
pst	pdt	yst	ydt	hst	hdt	gmt	bst	eet
eest	met	mest	wet	west	jst	aest	aesst	acst
acsst	awst	acdt	at	bt	btt	Cat	cct	cest
cet	ckt	clst	clt	cot	cut	ect	emt	fst
gst	gt	hfe	ict	ist	it	kdt	kst	lst
mdt	mpt	msd	msk	mt	mut	pmt	pnt	sst
tmt	tst	ut	wst	aedt	aft	ahdt	ahst	akdt
akst	amst	amt	anast	anat	art	azost	azst	azt
badt	bat	bdst	bdt	bet	bnt	bort	bot	bra
chadt	chast	chst	cxt	davt	ddut	dnt	dst	easst
east	eat	egst	egt	fdt	fjst	fjt	fkst	fkt
fwt	galt	gamt	gest	get	gft	gilt	gyt	haa
hac	hae	hap	har	hat	hay	hfh	hg	hkt
hna	hnc	hne	hnp	hnr	hnt	hny	hoe	idle
idlw	idt	iot	irdt	irkst	irkt	irst	irt	javt
jayt	jt	kgst	kgt	kost	krast	krat	lhdt	lhst
ligt	lint	lkt	magst	magt	mal	mart	mat	mawt
med	medst	mesz	mewt	mex	mez	mht	mmt	msks
mvt	myt	nct	ndt	nft	nor	novst	novt	npt
nrt	nsut	nt	nut	nzdt	nzst	nzt	oesz	oez
omsst	omst	pet	petst	pett	pgt	phot	pht	pkt
pmdt	pont	pwt	pyst	pyt	r1t	r2t	ret	rok

Table 1-2 Time Zones Supported by mailx (Continued)

sadt	sast	sbt	sct	set	sgt	srt	swt	tft
tha	that	tjt	tkt	tot	trut	tuc	tvt	ulast
ulat	usz1	usz1s	usz18	usz3	usz3s	usz4	usz4s	usz5
usz5s	usz6	usz6s	usz7	usz7s	usz8	usz8s	usz9	usz9s
utc	utz	uyt	uz10	uz11s	uz12s	uzt	vet	vlast
vlat	vtz	vut	wakt	wast	wat	wesz	wez	wft
wgst	wgt	wib	wita	wit	wtz	wut	yakst	yakt
yapt	yekst	yekt	azot	gz				

NOTE `mailx` displays an incorrect date if it reads an email message with the time zone information that is not listed in Table 1-2.

For more information about `mailx`, type `man 1M mailx` at the HP-UX prompt.

The mail/rmail Utility

You can use `mail`, the mail user agent to compose and send messages to users. The `mail` command, when used without arguments, displays all the messages, with the last received message displayed first. For each message, `mail` prints a `?` prompt, and reads a line from the standard input to determine the disposition of the message. `mail` exits automatically when the last message is displayed. It provides a set of command-line options to alter the messages being printed.

You can use the command `mail -e` to check for new mail messages. You can also edit the `mailfile` to alter the functioning of `mail`. For example, you can include the following line in `mailfile` to forward all mail addressed to the owner to a given machine or person:

```
Forward to <person>
```

This is used especially for forwarding mail to a given machine in a multiple-machine environment. The `Forward` option requires read-write group permission and `mail` group ID in the `mailfile`.

Unlike `mail`, you can use `rmail` only to send messages. `UUCP` uses `rmail` as a security precaution.

For more information on `mail` and `rmail`, type `man 1M mail` at the HP-UX prompt.

The Sendmail Utility

Sendmail acts as a post office, to which all messages can be submitted for routing. Sendmail interprets both Internet (that is, *user*@*domain*) and UUCP (that is, *host*!*user*) styles of addressing. The Sendmail configuration file controls how the addresses are interpreted. Sendmail can rewrite message addresses to conform to standards on many common target networks.

For more technical and conceptual information about `Sendmail`, HP recommends that you read *Sendmail, 3rd Edition*, by Bryan Costales with Eric Allman, published by O'Reilly and Associates, Inc. *Sendmail* discusses Sendmail Version 8.12, and some of its features may not be supported by Sendmail 8.11.1. You can also refer to the Sendmail 8.13 Companion by Bryan Costales. For information about using `Sendmail` with BIND, HP recommends that you read *DNS and BIND*, by Paul Albitz and Cricket Liu, also published by O'Reilly and Associates, Inc.

You can get information about the O'Reilly books (availability, how to order them, and so on) by visiting the O'Reilly Website:

`http://www.ora.com`

You also can visit the Website for Sendmail:

`http://www.sendmail.org`

NOTE

Sendmail 8.11.1 for HP-UX 11i v2 is an HP implementation of publicly available Sendmail 8.11.1. HP provides support for the features documented in this chapter and in the `sendmail`(1M) manpage.

All occurrences of the term Sendmail in this chapter and "Configuring and Administering Sendmail" on page 39 refer to Sendmail 8.11.1.

This section discusses the following topics:

- "Default Client/Server Operation" on page 35
- "How Sendmail Handles Errors" on page 36

Message Structure

A message has three parts: an envelope, a message header, and a message body.

The **envelope** consists of the sender address, recipient address, and routing information shared by programs that create, route, and deliver the message. It is usually not seen directly by either the sender or the recipients of the message.

The **message header** consists of a series of standard text lines used to incorporate address, routing, date, and other information into the message. Header lines may be part of the original message and may also be added or modified by the various mail programs that process the message. Header lines may or may not be used by these programs as envelope information.

By default, the first blank line in the message terminates the message header. Everything that follows is the **message body** and is passed uninterpreted from the sender to the recipient.

How Sendmail Collects Messages

Sendmail receives messages through any of the following methods:

- A user agent calls Sendmail to route a piece of mail. User agents supported by HP for use with Sendmail are `elm`, `mail`, and `mailx`.
- A Sendmail daemon or other mail program calls Sendmail to route a piece of mail received from the network or the mail queue.
- A user invokes Sendmail directly from the command line.

How Sendmail Routes Messages

Sendmail routes messages as follows:

1. Rewrites the recipient and sender addresses given to it, to comply with the standards of the target network.
2. If necessary, adds lines to the message header to enable the recipient to reply.

3. Passes the mail to one of the several specialized delivery agents for delivery.

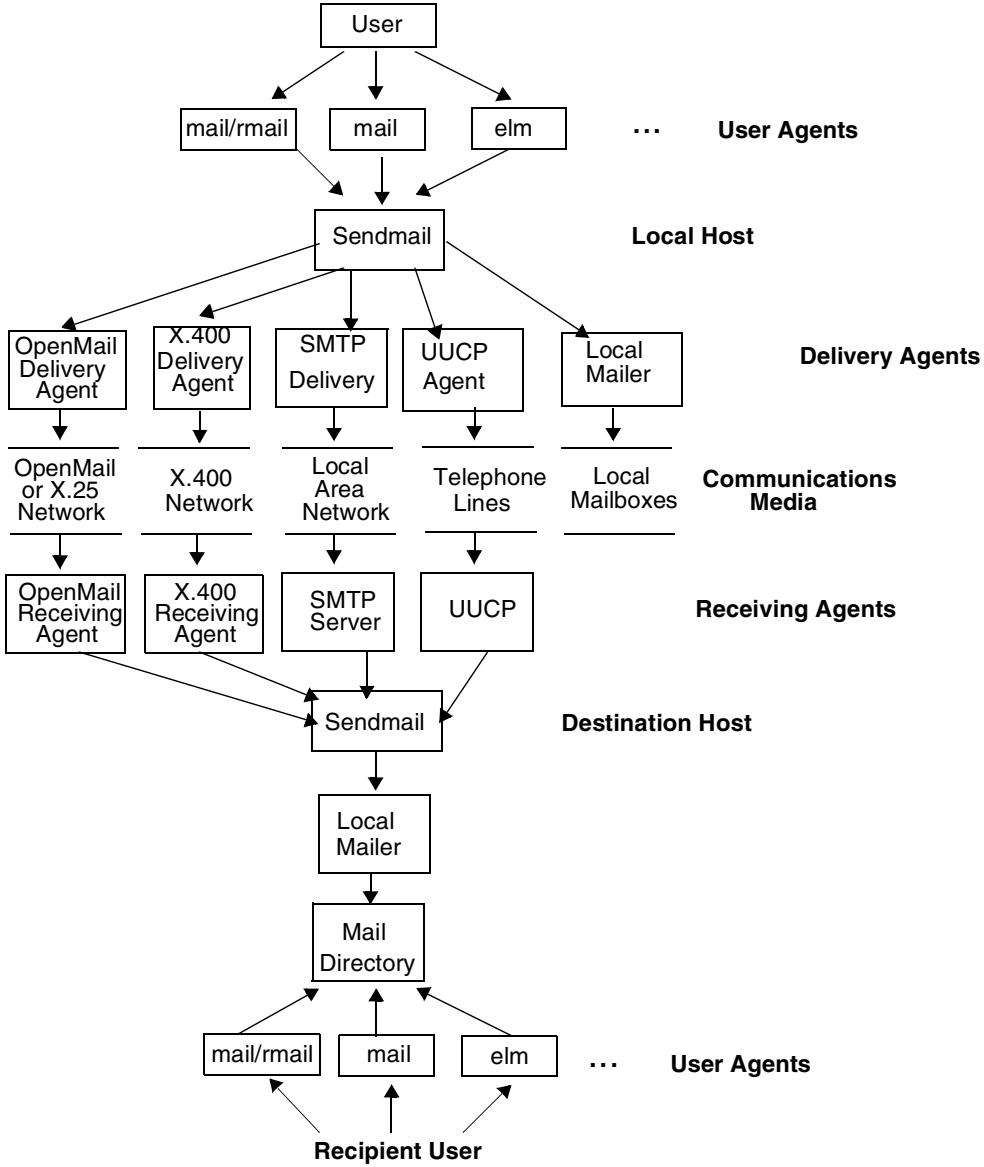
Figure 1-1 outlines the flow of messages through Sendmail.

After Sendmail collects a message, it routes the message to each of the specified recipient addresses. In order to route a message to a particular address, Sendmail must resolve that address to a {*delivery agent*, *host*, *user*} triple. This resolution is based on the rules defined in the Sendmail configuration file, `/etc/mail/sendmail.cf`.

Sendmail invokes a separate delivery agent for each host to which messages are being routed. Some delivery agents can accept multiple users in a given invocation. Others must be invoked separately for each recipient. Delivery agents that HP supports for use with Sendmail include SMTP, UUCP, X.400, and OpenMail.

To invoke a delivery agent, Sendmail constructs a command line according to a template in the configuration file. If the delivery agent is specified as IPC, Sendmail does not invoke an external delivery agent. Instead, Sendmail opens a TCP/IP connection to the SMTP server on the specified host and transmits the message using SMTP.

Figure 1-1 Flow of Mail Through Sendmail



If an address resolves to the local mailer, Sendmail looks up the address in its alias database and expands it appropriately if it is found. The aliasing facility or a user's `.forward` file can be used to route mail to programs and to files. (Sendmail does not mail directly to programs or files.) Mail to programs is normally piped to the `prog` mailer (`/usr/bin/sh -c`), which executes the command specified in the alias or `.forward` file definition. (You can restrict the programs that can be run through the `aliases` or `.forward` files. See "Security" on page 74 for more information.) Mail to a file is directly appended to the file by Sendmail if certain conditions of ownership and permission are met.

After expanding all the aliases, Sendmail routes mail that is addressed to a local user to the local mailer (`/usr/bin/rmail`), which deposits the message in the user's mailbox.

Default Routing Configuration

The installed configuration file, if unmodified, routes mail depending on the syntax of the recipient addresses as described in the following sections.

Local Addresses: The following forms are recognized as local addresses and are delivered locally:

```
user
user@localhost
user@localhost.localdomain
user@alias
user@alias.localdomain
user@[local_host's_internet_address]
localhost!user
localhost!localhost!user
user@localhost.uucp
```

UUCP Addresses: Addresses of the following forms are recognized as UUCP addresses, where *`host`* is not the local host name:

```
host!user
host!host!user
user@host.uucp
```

If your host has a direct UUCP connection to the next host in the path, the mail is delivered to that host through UUCP. If not, the message is returned with an error. The supplied configuration file provides detailed instructions for arranging to relay such mail through hosts to which you can connect.

SMTP Addresses: RFC 2822-style addresses in any of the following forms, where *host* is not the local host name, are routed by SMTP over TCP/IP:

```
user@host
user@host.domain
<@host,@host2,@host3:user@host4>
user@[remote_host's_internet_address]
```

If the name server is in use, Sendmail requests mail exchanger (MX) records for the remote host. If there are any, it attempts to deliver the mail to each of them, in the order of preference, until delivery succeeds.

Otherwise, Sendmail connects directly to the recipient host and delivers the message.

Mixed Addresses: The supplied configuration file interprets address operators with the following precedence:

```
@, !, %
```

This means that recipient addresses using mixtures of these operators are resolved as shown in Table 1-3.

Table 1-3 How Sendmail Resolves Addresses with Mixed Operators

Address	Mailer	Host	User	Recipient
user%hostA@hostB	TCP	hostB	user%hostA @hostB	user@hostA
user!hostA@hostB	TCP	hostB	hostA!user @hostB	hostA!user
hostA!user%hostB	UUCP	hostA	user@hostB	user@hostB

Mail Exchanger (MX) Records

The BIND name server, if it is in use on your host, provides MX records. These can be used to notify Sendmail that mail for a particular host can be relayed by another host, if the addressed host is temporarily down or

otherwise inaccessible. For information on creating MX records, see *HP-UX IP address and Client Management Guide* at the URL http://www.docs.hp.com/hpux/netcom/index.html#Internet%20Services.

MX records are used only if a message address resolves to an IPC mailer (that is, one that uses SMTP over sockets to perform delivery). Instead of attempting to connect directly to the recipient host, Sendmail first queries the name server, if it is running, for MX records for that host. If the name server returns any answer, Sendmail sorts them in preference order, highest preference (lowest number) first. If the local host appears in the list, the local host and any MX hosts with lower preference (higher numbers) are removed from the list. If any MX hosts remain, Sendmail then tries to connect to each MX host in the list in order, and it delivers the message to the first MX host to which it successfully connects. If that MX host is not the final destination for the message, it is expected that the host will relay the message to its final destination.

If Sendmail tries all the MX hosts in the list and fails, the message is returned to the sender with an error message. If you want Sendmail to try to connect to the host to which the message is addressed, uncomment the following line in the /etc/mail/sendmail.cf file:

```
TryNullMXList
```

Sendmail then tries to connect to the host to which the message is addressed, if any of the following conditions occur:

- The name server returns no MX records.
- The name server is not running.
- The local host is the highest preference mail exchanger in the list.

At log level 11 and above, Sendmail logs in the system log the name and Internet address of the MX host (if any) to which it delivered (or attempted to deliver) a message.

MX records are used for two main purposes:

- To arrange one host backup by receiving mail for the host when it is down
- To arrange the mail addressed to remote networks be relayed through the appropriate gateways

In the following example, the name server serving the domain paf.edu has the following MX records configured to provide backup for host bling:

```
;name  ttl  class   MX   preference    mail exchanger
bling       IN      MX   0             bling.paf.edu.
            IN      MX   20            wheo.paf.edu.
            IN       MX   30             munch.pag.edu.
```

Normally, mail for `bling` will go directly to `bling`. However, if `bling` is down, or if the sending host cannot connect to `bling`, Sendmail will route mail for `bling` to `wheo`. If `wheo` is also down or unreachable, Sendmail will route the mail to `munch`. Naturally, for this to be useful, `wheo` and `munch` must be able to route mail to `bling`.

Assuming that the host and its mail exchangers see the same `MX` data from the name server, each host that has `MX` records should have an `MX` record for itself, and the preference on its own record should be the highest (that is, the lowest number) in the list.

The following example relays messages through a gateway:

```
;name   ttl   class   MX  preference    mail exchanger

*.nz.         IN      MX  0             gw.dcc.nz.
```

Messages addressed to hosts in the `nz` domain are relayed to the host `gw.dcc.nz`. HP recommends that you seek permission from the administrators of hosts not under your own control before relaying mail through them.

MX Failures: Several possible failures are associated with `MX` configuration:

- The name server query for `MX` records fails.

  The query fails because no `MX` records exist for the target host or because the name server is not running. You can set the `TryNullMXList` option in the `/etc/mail/sendmail.cf` file if you want Sendmail to always try to connect to the host to which the message is addressed.

  If the query fails temporarily (that is, `h_errno` is set to `TRY_AGAIN`) the message is queued. The possible values of `h_errno` are documented in the header file `/usr/include/netdb.h`.

- Connection attempts to the hosts in the `MX` list all fail.

  Sendmail reports the failure attempting to connect to the last `MX` host (that is, the highest preference value) in the list that it tried. For example, with mail exchangers configured as in the `paf.edu` example earlier, if the attempts to connect to `bling` and `wheo` result

in temporary failures, but the attempt to connect to `munch` fails permanently, the message is returned as an error. If the attempts to connect to `bling` and `wheo` result in permanent failures, but the attempt to connect to `munch` fails temporarily, the message is queued.

- A host cannot deliver a message to another host for which it is a mail exchanger.

  This failure is handled as a normal delivery failure, either by the mail exchanger host or by the host sending to the mail exchanger.

How Sendmail Improves Mail Queue Performance

Mail queue performance is impacted by the number of entries in the queue directories. Multiple Queue Directories improves mail queue performance in Sendmail. This feature facilitates the parallel processing of mail by spreading process loads across multiple disks, thereby improving the queue performance. UNIX files take a long time to open when entries in the directories exceed 100.

In order to use multiple directories, you must supply the `QueueDirectory` option in the `sendmail.cf` file with a value ending with *.

For example, if you specify the following in the configuration file, all the directories or links to directories that begin with `g` will be used:

```
O QueueDirectory=/var/spool/mqueue/g*
```

If there are five directories, `g1`, `g2`, `g3`, `g4`, and `g5`, Sendmail uses all five directories when the Sendmail daemon is restarted. Mail is randomly assigned to the queue directories. Do not change the queue directory structure when Sendmail is running.

You can flush individual mail queues by specifying the following on the command line:

```
sendmail -q -O QueueDirectory=/var/spool/mqueue/g1
```

```
sendmail -q -O QueueDirectory=/var/spool/mqueue/g3
```

You can use the `mailq` command to display the mail queue, as shown in the following example:

```
#mailq
   /var/spool/mqueue/g1 is empty
   /var/spool/mqueue/g2 (1 request)
--Q-ID--- -Size- --Q-Time-- ----------Sender/Recipient--
```

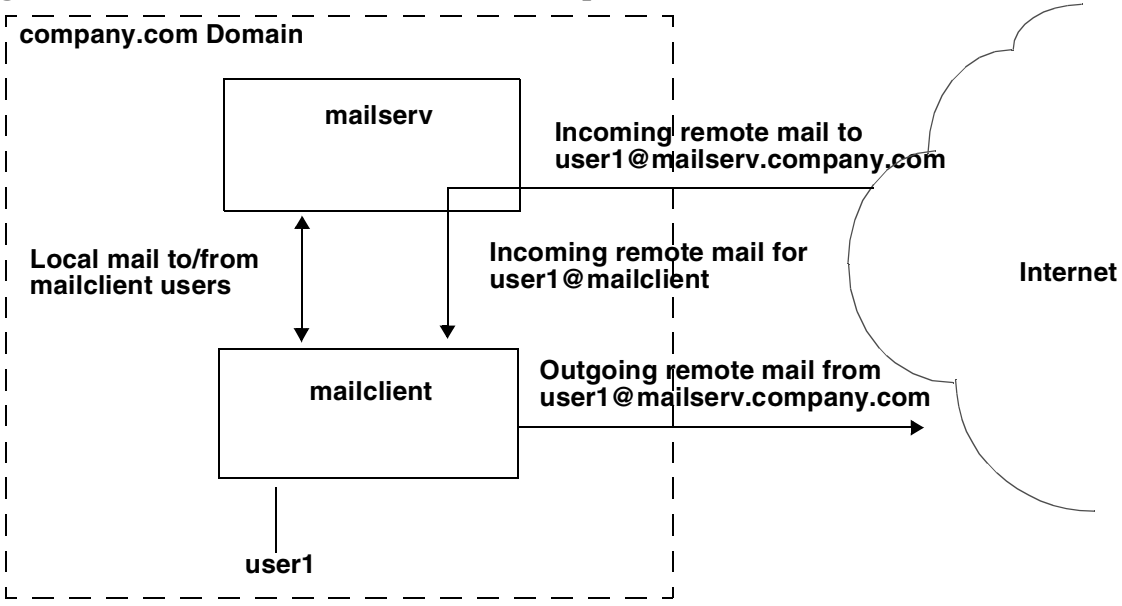
```
gBJ2va   02544    5 Wed Dec 18 21:57   root root
  /var/spool/mqueue/g2 is empty
  /var/spool/mqueue/g3 is empty
Total Requests: 0
```

An efficient queue file-naming system is also being provided in this release. The algorithm used to name files ensures that the names will be unique for 60 years. The queued items can be moved between queues with ease.

Default Client/Server Operation

This section describes the operation of Sendmail servers and clients. Figure 1-2 shows a Sendmail server called `mailserv` and a Sendmail client called `mailclient` in the `company.com` domain. On `mailclient`, the `SENDMAIL_SERVER_NAME` in the `/etc/rc.config.d/mailservs` file is set to `mailserv.company.com`. `user1` is a user on `mailclient`.

Figure 1-2 Sendmail Client-Server Operation



Outgoing mail from `user1` can be local mail that is intended for any user on `mailclient`. Local mail is forwarded to `mailserv`; you specify this by setting the `DH` macro entry in the `/etc/mail/sendmail.cf` file on

mailclient. (The Sendmail installation script sets the DH macro value to the host specified by SENDMAIL_SERVER_NAME.) Outgoing mail that is not local is sent by mailclient to the remote host using MX records. Because the DM macro entry in the /etc/mail/sendmail.cf file on mailclient is set to mailserv.company.com, mail from user1 appears to be from user1@mailserv.company.com.

Because mail sent to remote hosts from user1 is sent from user1@mailserv.company.com, replies to user1's messages are returned to mailserv. On mailserv, when Sendmail receives mail for user1, it looks up user1 in the aliases database and redirects mail for user1 to user1@mailclient.

You can modify Sendmail server and client operations. Most modifications involve changing or re-creating the /etc/mail/sendmail.cf file on the server or client systems. For example, you can define the DM macro on a mail server system. You can also modify the /etc/mail/sendmail.cf file so that the clients relay all outbound mail to the server; this is described in "Modifying the Default Sendmail Configuration File" on page 48.

How Sendmail Handles Errors

By default, Sendmail immediately reports to standard output any errors that occur during the routing or delivery of a message. Sendmail distinguishes between **temporary failures** and **permanent failures**.

Permanent failures are mail transactions that are unlikely to succeed without the intervention of the sender or a system administrator. For example, mailing to an unknown user is a permanent failure. A delivery failure of the local mailer because the file system is full is also a permanent failure.

Temporary failures are mail transactions that might succeed if retried later. For example, an error message connection refused displayed while attempting to connect to a remote SMTP server is a temporary failure, since it probably means that the server is temporarily not running on the remote host.

How Sendmail Handles Permanent Failures

Permanent failures include the following:

- Temporary failures that have remained in the mail queue for the queue timeout period (set with the `Timeout.queuereturn` option in the `/etc/mail/sendmail.cf` file), which is normally five days.
- Local recipient user unknown.
- The recipient address cannot be resolved by the configuration file.
- Permanent delivery agent (mailer) failures.
- Inability to find an Internet address for a remote host.
- A remote SMTP server reports an address is undeliverable during the SMTP transaction.

In most cases, if message delivery fails permanently on a remote system, mail that includes a transcript of the failed delivery attempt and the undelivered message is returned to the sender. This transcript includes any standard error output from the delivery agent that failed.

If Sendmail tries all `MX` hosts in its preference list and fails to deliver a message, the message is returned to the sender with an error message. For more information, see "Mail Exchanger (MX) Records" on page 31.

If delivery failed on an alias, and an owner is configured for that alias in the aliases database, Sendmail returns the message and transcript to the alias owner.

If the message header contains an `Errors-To:` header line, Sendmail returns the message and transcript to the address on the `Errors-To:` line instead of to the sender's address.

If the Postmaster Copy option (option `P`) is set to a valid address, Sendmail sends a copy of the transcript and failed message (with the message body deleted) to the Postmaster Copy address.

If the attempt to return the failed message itself fails, Sendmail returns the message and transcript to the alias `postmaster` on the local system. The `postmaster` alias in the default alias file (`/usr/newconfig/etc/mail/aliases`) resolves to root.

If Sendmail is unable to return the message to any of the addresses described previously, as a last resort it appends the error transcript and returned message to the file `/var/tmp/dead.letter`.

Finally, if this fails, Sendmail logs the failure and leaves the original failed message in the mail queue so that a future queue-processing daemon will try to send it. If this fails, an error message is returned again.

How Sendmail Handles Temporary Failures

Messages that fail temporarily are saved in the mail queue and retried later. By default, the mail queue is stored in the directory `/var/spool/mqueue`. Sendmail saves the message components in two files created in the mail queue directory. The message body is saved in a data file, and the envelope information, the header lines, and the name of the data file are saved in a queue control file.

Typically, the Sendmail daemon is run with the `-q` *`time_interval`* option, as in the following example:

```
/usr/sbin/sendmail -bd -q30m
```

In this example, every 30 minutes, Sendmail processes any messages currently in the queue.

While processing the queue, Sendmail first creates and sorts a list of the messages in the queue. Sendmail reads the queue control file for each message to collect the preprocessed envelope information, the header lines, and the name of the data file containing the message body. Sendmail then processes the message just as it did when it was originally collected.

If Sendmail detects, from the time stamp in a queued message, that the message has been in the mail queue longer than the queue timeout, it returns the message to the sender. The queue timeout is set with the `Timeout.queuereturn` option in the `/etc/mail/sendmail.cf` file and, by default, is five days.

2 Configuring and Administering Sendmail

This chapter describes Sendmail, the Internet Services mail routing utility provided on the HP-UX operating system. Sendmail relays

incoming and outgoing mail messages to the appropriate programs for delivery and further routing. Sendmail allows you to send mail and to receive mail messages from other hosts on a local area network or through a gateway.

This chapter contains the following sections:

NOTE You cannot use the System Administration Manager (SAM) to install, configure, or enable Sendmail on the HP-UX operating system.

Configuring Sendmail

Sendmail is packaged with the core HP-UX 11i v2 operating system. When you install the operating system, Sendmail is automatically installed on your system. The necessary files required for Sendmail operation are created or modified on your system. The Sendmail configuration file supplied with the operating system, `sendmail.cf`, will work without modifications for most installations.

Therefore, you only need to perform a few tasks to configure Sendmail:

- Set up Sendmail servers to run with NFS.
- Configure and start Sendmail clients.
- Verify that Sendmail is running properly.

This section discusses the following topics:

NOTE HP recommends that you use Sendmail with the BIND name server. The BIND name server must have a mail exchanger (`MX`) record for every host in every domain that it serves. For more information on how Sendmail uses `MX` records, see "Mail Exchanger (MX) Records" on page 31.

Configuring Sendmail on a Standalone System

When Sendmail is installed, it is automatically configured to send and receive mail messages for users on the local system only. The standalone system processes all outbound mail and establishes connections to the message destination host or to the `MX` hosts. Because the Sendmail daemon is invoked automatically when a system is rebooted, no system files need to be modified.

The installation script makes the following configuration changes:

- Sets the `SENDMAIL_SERVER` variable in the `/etc/rc.config.d/mailservs` file to 1. This ensures that the Sendmail daemon is started whenever you reboot your system or run the Sendmail startup script.
- Creates `/etc/mail/sendmail.cf` and `/etc/mail/aliases` files with default configurations. These files are created with `root` as the owner and `other` as the group. The permission for `/etc/mail/aliases` and `/etc/mail/sendmail.cf` is set to 0640 and 0444, respectively.

NOTE

If the `/etc/mail/sendmail.cf` file already exists, the existing file is saved to `/etc/mail/#sendmail`. If the `/etc/mail/aliases` file already exists, the Sendmail installation script does not recreate the `aliases` file.

- Creates the `/etc/mail/sendmail.cw` file that contains the host name and the fully qualified host name for the system. For example, the system `dog` in the domain `hp.com` contains the following entries in the `sendmail.cw` file:

  ```
  dog

  dog.hp.com
  ```

- Finally, the installation script issues the following command to run the Sendmail startup script:

  ```
  /sbin/init.d/sendmail start
  ```

  The Sendmail startup script generates the aliases database from the `/etc/mail/aliases` source file. The generated database is located in the `/etc/mail/aliases.db` file.

  The Sendmail startup script then invokes the Sendmail daemon by issuing the following command:

  ```
  /usr/sbin/sendmail  -bd  -q30m
  ```

  By using the `-q30m` option, Sendmail processes the mail queue every 30 minutes.

  For more information about Sendmail's command line options, type `man 1M sendmail` at the HP-UX prompt.

Configuring Sendmail on a Mail Server

This section describes how to configure a system to allow users on other (client) systems to use Sendmail.

The mail server receives mail for local users and for the users on client systems. Users on client systems mount the mail directory from the server and read or access mail over an NFS link. For more information on how Sendmail clients and servers work, see "Default Client/Server Operation" on page 35.

The Sendmail installation script performs the configuration changes that are described in "Configuring Sendmail on a Standalone System" on page 41. To set up the system as an NFS server and allow the Sendmail clients to read and write to the `/var/mail` directory, do the following:

1. Ensure that all mail users have accounts on the mail server and that their user IDs and group IDs on the mail server are the same as on the client machines. (This step is not necessary if you are using NIS or NIS+ and your mail server is in the same NIS or NIS+ domain as the clients.)
2. Use a text editor to set the `NFS_SERVER` variable to 1 in the `/etc/rc.config.d/nfsconf` file.
3. Use a text editor to add the following line to the `/etc/exports` file:

   ```
   /var/mail  -access=client1,client2, ...
   ```

   where each mail client is listed in the access list. If the `/etc/exports` file does not exist, you must create it.
4. Issue the following command to run the NFS startup script:

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

For more information on NFS, see *Installing and Administering NFS Services*, at the URL `http://www.docs.hp.com/hpux/onlinedocs/B1031-90048/B1031-90048.html`.

Configuring Sendmail on a Mail Client

Sendmail clients do not receive mail on their local system, but receive mail on the mail server. User mail directories reside on the server, and users read their mail over an NFS link. By default, a Sendmail client forwards to the server any local mail (a user address destined for the

client system) and sends nonlocal mail directly to the destination system or MX host. An outgoing mail message appears to originate from the server, so replies are sent back to the server. For more information on how Sendmail clients and servers work, see "Default Client/Server Operation" on page 35. Sendmail clients can be diskless systems.

To configure a Sendmail client system, do the following:

1. Use a text editor to set the SENDMAIL_SERVER variable to 0 in the /etc/rc.config.d/mailservs file. This ensures that the Sendmail daemon will not be started when you reboot your system or run the Sendmail startup script.
2. Set the SENDMAIL_SERVER_NAME variable in the /etc/rc.config.d/mailservs file to the host name or to the IP address of the mail server you will use (the machine that will run the Sendmail daemon).
3. Set the NFS_CLIENT variable to 1 in the /etc/rc.config.d/nfsconf file.
4. Add the following line in the /etc/fstab file:

   ```
   servername:/var/mail  /var/mail  nfs  0  0
   ```

   where servername is the name configured in the SENDMAIL_SERVER_NAME variable in /etc/rc.config.d/mailservs. If the /etc/fstab file does not exist, you must create it.
5. Issue the following command to run the Sendmail startup script:

   ```
   /sbin/init.d/sendmail start
   ```
6. Issue the following command to run the NFS startup script:

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

The Sendmail startup script assumes that this system will use the host specified by the SENDMAIL_SERVER_NAME variable as the mail hub. The script also assumes that mail sent from this system appears to be from the host specified by the SENDMAIL_SERVER_NAME variable (this feature may previously have been known as **site hiding**). The script therefore modifies the macros DM (for **masquerade**) and DH (for **mail hub**) in the system's /etc/mail/sendmail.cf file to use the host specified by the SENDMAIL_SERVER_NAME variable. If the DM and DH macros have been defined previously, the startup script does not modify them.

The client system now forwards local mail to the mail server and forwards other mail directly to remote systems. To configure the client system to relay all mail to the mail server for delivery, see "Modifying the Default Sendmail Configuration File" on page 48.

The NFS startup script `mounts` the `/var/mail` directory from the mail server to your system.

Verifying your Sendmail Installation

This section provides information on how to verify your Sendmail installation. It discusses the following topics:

Sending Mail to a Local User

To check your local mailer or user agent, send a mail message to a local user (for example, `joe`) on your system:

```
date  |  mailx  -s  "Local sendmail Test"  joe
```

This must result in a message similar to the following being sent to user `joe`:

```
From joe Wed Aug 6 09:18 MDT 2002
Received: by node2; Wed, 6 Aug 02 09:18:53 mdt
Date: Wed, 6 Aug 02 09:18:53 mdt
From: Joe User <joe>
Return-Path: <joe>
To: joe
Subject: Local sendmail Test

Wed Aug 6 09:18:49 MDT 2002
```

An entry in your `/var/adm/syslog/mail.log` file must have been logged for the local message transaction. See "Configuring and Reading the Sendmail Log" on page 97 for more information.

Using UUCP Addressing to Send Mail to a Remote User

If you are using UUCP addressing, you can verify your Sendmail installation by sending a mail message to a remote user with UUCP transport by using a *host*!*user* address, where *host* is a system to which your local host has a direct UUCP connection. (The `uuname` command lists the UUCP names of known systems. Type `man 1 uuname` at the HP-UX prompt for more information.)

To verify both inbound and outbound UUCP connections, mail the message in a loop, using the syntax *remote_host*!*my_host*!*user*. For example, if you execute the following command:

```
date | mailx -s "UUCP Test" node1!node2!joe
and node2 is your local host, you must receive a message simil
ar to this:
From node1!node2!joe Wed Aug  6 09:48 MDT 2003
Received: by node2; Wed, 6 Aug 02 09:48:09 mdt
Return-Path: <node1!node2!joe>
Received: from node1.UUCP; Wed, 6 Aug 02 09:30:16
Received: by node1; Wed, 6 Aug 02 09:30:16 mdt
Received: from node2.UUCP; Wed, 6 Aug 02 09:26:18
Received: by node2; Wed, 6 Aug 02 09:26:18 mdt
Date: Wed, 6 Aug 02 09:26:18 mdt
From: Joe User <node1!node2!joe>
To: node1!node2!joe
Subject: UUCP Test

Wed Aug  6 09:26:15 MDT 2002
```

An entry in your `/var/adm/syslog/mail.log` file must have been logged for the UUCP mail transaction. See "Configuring and Reading the Sendmail Log" on page 97 for more information.

NOTE

In this example, if you send a mail message to yourself and if the remote system is running Sendmail, ensure that the `MeToo` option is set in the configuration file on the remote system. The remote system's configuration file must contain a line beginning with `O MeToo`. If the remote host's configuration file does not contain such an entry, Sendmail on the remote host notices that the sender is the same as the recipient and removes your address from the recipients' list.

Using SMTP Transport to Send Mail to a Remote User

If you are using the SMTP Transport, you can verify your Sendmail installation by sending a message to a remote user using a *user@host* address, where *host* is a system that provides an SMTP server (for example, the Sendmail daemon).

To verify both inbound and outbound SMTP connections, mail the message in a loop, using the syntax *user%my_host@remote_host*. For example, if you try:

```
lx -s "Round Robin SMTP" joe%node2@node1
you must receive a message similar to the following:
From joe@node2 Wed Aug  6 14:22 MDT 2003
Received: from node1 by node2; Wed, 6 Aug 02 14:22:56  mdt
Return-Path: <joe@node2>
Received: from node2 by node1; Wed, 6 Aug 02 14:25:04  mdt
Received: by node2; Wed, 6 Aug 02 14:22:31 mdt
Date: Wed, 6 Aug 02 14:22:31 mdt
From: Joe User <joe@node2>
To: joe%node2@node1
Subject: Round Robin SMTP

Wed Aug  6 14:22:28 MDT 2002
```

An entry in your `/var/adm/syslog/mail.log` file must have been logged for the SMTP mail transaction. See "Configuring and Reading the Sendmail Log" on page 97 for more information.

NOTE

In this example, if you send a mail message to yourself and if the remote system is running Sendmail, ensure that the `MeToo` option is set in the configuration file on the remote system. The remote system's configuration file must contain a line beginning with `O MeToo`. If the remote host's configuration file does not contain such an entry, Sendmail on the remote host notices that the sender is the same as the recipient and removes your address from the recipients' list.

Modifying the Default Sendmail Configuration File

The Sendmail configuration file that is supplied with HP-UX works correctly for most Sendmail configurations, so you probably do not need to modify the configuration file. However, certain modifications to the file are supported. This section describes examples of modifications that you may want to make. The configuration file also contains instructions for making the supported modifications.

This section discusses the following topics:

CAUTION HP supports the default configuration file and all the modifications described in it. If you make any changes other than the ones described in the default configuration file, HP cannot support your configuration.

The Sendmail Configuration File

The Sendmail configuration file, `/etc/mail/sendmail.cf`, performs the following functions:

- Defines certain names and formats, such as the name of the sender for error messages (`MAILER-DAEMON`), the banner displayed by the SMTP server on startup, and the default header field formats.
- Sets values of operational parameters, such as timeout values and logging level.
- Specifies how mail will be routed. In other words, it specifies how recipient addresses are to be interpreted.
- Defines the delivery agents (mailers) to be used for delivering the mail.

- Specifies how Sendmail must rewrite addresses in the header, if necessary, so that the message address can be understood by the receiving host. The address rewriting process is controlled by sets of address rewriting rules called **rulesets**.

The default configuration file, `sendmail.cf`, is located in the `/usr/newconfig/etc/mail/sendmail.cf` directory, and is installed in the `/etc/mail/sendmail.cf` directory.

HP recommends that you leave a copy of the configuration file in the `/usr/newconfig` directory unmodified, in case you need to reinstall the default configuration settings.

To modify the configuration settings in the `/etc/mail/sendmail.cf` file, perform the following steps:

1. The `gen_cf` UNIX shell script is installed in the `/usr/newconfig/etc/mail/cf/cf` directory. You cannot copy this script to a different directory and execute it, because it uses the macros defined in the `/usr/newconfig/etc/mail/cf` directory to generate the `sendmail.cf` file.

   This script provides many options that enable a specific ruleset. The `*.m4` files defined in the `/usr/newconfig/etc/mail/cf` directory are the input files for this script. You can specify the output file, and later incorporate site-specific changes (if any) in the output file.

   Run the script `gen_cf` from the HP-UX prompt. A list of options that enable a specific ruleset is displayed.

2. Choose the appropriate option. See “Sendmail Configuration Options” on page 50 for a description of options.

   An updated configuration file, `sendmail.cf.gen`, is generated in the directory `/usr/newconfig/etc/mail/cf/cf`.

3. Copy or move the `sendmail.cf.gen` file to `/etc/mail` directory as `sendmail.cf`. After copying the `sendmail.cf.gen` file to the `/etc/mail` directory, you can make certain site-specific modifications to the sendmail.cf file.

   If you do not wish to generate the sendmail.cf file using the gen_cf script, you can directly make modifications to the `/etc/mail/sendmail.cf` file.

Restarting Sendmail

Issue the following commands, on a standalone system or on the mail server, to restart Sendmail:

- `/sbin/init.d/sendmail stop`

  `/sbin/init.d/sendmail start`

You must restart Sendmail if changes are made to any of the following:

- The Sendmail configuration file, `/etc/mail/sendmail.cf`.
- The UUCP configuration, as reflected in the output of the `uuname` command.

Sendmail Configuration Options

This section describes Sendmail configuration options.

Maximum message size (option MaxMessageSize)

This option restricts the maximum message (in bytes) that sendmail will accept from a remote system. If a message larger that this limit is originated form the local system, the message will be truncated to the limit.

To enable this feature uncomment the line:

O MaxMessageSize=100000

Forwarding Nondomain Mail to a Gateway

Mail that is being sent to a domain other than the sender's domain can be forwarded to a mail gateway. To have nondomain mail forwarded to a mail gateway, edit the DS line in the `/etc/mail/sendmail.cf` file to specify the host name of the mail gateway:

```
DSmailgw.hp.com
```

Setting Mail Header Lengths

You can set a limit for the mail header. The maximum header length by default is 32768. To change the mail header length:

1. Open the `sendmail.cf` file.

2. Set the value of the option `MaxHeadersLength=n`, where `n` is the maximum number of lines allowed in the mail header.

If a mail header exceeds the maximum value, the following error message is displayed to the sender:

```
552 Headers too larger #MaxHeadersLength
```

Limiting Message Recipients

By default, the maximum number of recipients is 100. You can limit the number of users allowed to receive a single mail message. This helps to prevent the flow of spam on the mail server.

- In the `sendmail.cf` file, set the value of `MaxRecipientsPerMessage=n`, where `n` is the maximum number of recipients allowed for a single mail message.

After a message has been sent to the maximum number of recipients allowed, Sendmail sends the error message `452 Too many recipients` to the sender of the message.

This will work only when all the recipients of the mail message have their mailboxes on the same machine.

Timeout.*

- You can set the total time spent in satisfying a socket control request using the `Timeout.control` option. The default setting for this option is:

  ```
  #O Timeout.control=2m
  ```

- You can set the resolver's transmission time interval (in seconds) using the `Timeout.resolver.retrans` option. This option sets the `Timeout.resolver.retrans.first`, which sets the resolver's transmission time interval (in seconds) for the first attempt to deliver a message. It also sets the `Timeout.resolver.retrans.normal` option. The default setting for this option is:

  ```
  #O Timeout.resolver.retrans=5s

  #O Timeout.resolver.retrans.first=5s

  #O Timeout.resolver.retrans.normal=5s
  ```

- You can set the frequency of resolver query retransmission using the `Timeout.resolver.retrans.normal` option. This option sets the `Timeout.resolver.retry.first` option for the first attempt to deliver a message. It also sets the `Timeout.resolver.retry.normal` option for all resolver lookups except for the first delivery attempt. The default setting for this option is:

  ```
  #O Timeout.resolver.retry=4

  #O Timeout.resolver.retry.first=4

  #O Timeout.resolver.retry.normal=4
  ```

DataFileBufferSize

Use this option to control the maximum size of a memory-buffered data (df) file before using a disk-based file. The default setting for this option is:

```
#O DataFileBufferSize=4096
```

XscriptFileBufferSize

Use this option to control the maximum size of a memory-buffered (xf) transcript before using a disk-based file. The default setting for this option is:

```
#O XscriptFileBufferSize=4096
```

MaxAliasRecursion

You can specify the maximum depth of an alias recursion in the `sendmail.cf` file using this option. The default setting for this option is:

```
#O MaxAliasRecursion=10
```

PidFile

You can define the location of the ProcessId (`Pid`) file using this option. The default setting for this option is:

```
#O PidFile=/etc/mail/sendmail.pid
```

`/etc/mail/sendmail.pid` is taken as the default file if this option is not set. If you choose a directory other than `/etc/mail` for the `pid` file, ensure that the directory has the same write permissions as those of `/etc/mail`.

ProcessTitlePrefix

You can specify the prefix string for the process title shown in the `ps` listings using this option. By default, this option is commented. For example, if you set this option in the `sendmail.cf` file as:

```
O ProcessTitlePrefix=HPUX_Sendmail-8.11.1
```

the command `ps -ef | grep sendmail | grep -v grep` displays `sendmail: accepting connections` in the output.

TrustedUser

You can use this option to specify a user who can own important files instead of root. This option necessitates `fchown`. The default setting for this option is:

```
#O TrustedUser=root
```

MaxMimeHeaderLength

You can set the size of the MIME headers and parameters within those headers using this option. You can also use this to protect Mail User Agents (MUA) from buffer overflow attacks. The default setting for this option is unlimited, as shown in the following example:

```
#O MaxMimeHeaderLength=0/0
```

DeadLetterDrop

Use this option to specify the location of the system-wide `dead.letter` file, which was formerly hardcoded to `/var/tmp/dead.letter`. The default setting for this option in this version is:

```
O DeadLetterDrop=/var/tmp/dead.letter
```

Sendmail does not save mail anywhere if this option is not set.

Options Configured Using the /usr/newconfig/etc/mail/cf/cf/gen_cf Script

Following are the options that you can configure in Sendmail using the `/usr/newconfig/etc/mail/cf/cf/gen_cf` script:

NOTE When you create a new `sendmail.cf` file using the `gen_cf` script, the new configuration file does not contain any change that you have added directly to the `sendmail.cf` file. You must reapply any such change to

the newly created configuration file. Therefore, HP recommends that you take backup of the configuration file that contains your changes, in case you want to run the `gen_cf` script again to generate the configuration file again.

Relay On This option is equivalent to selecting the following `/usr/newconfig/etc/mail/cf/cf/gen_cf` script options while generating the `/usr/newconfig/etc/mail/cf/cf/sendmail.cf.gen` file:

- Accept unresolvable domains
- Accept unqualified senders
- Promiscuous relay

Relay OFF This option generates a `sendmail.cf` file which is identical to the default `sendmail.cf` available in the `/usr/newconfig/etc/mail/` directory.

If this option is used with mutually exclusive options, this option does not turn OFF the relay. The other options take precedence over the RELAY OFF option.

Relay Entire Domain Setting this option, will allow any host in your domain as defined by the `m` class macro (`$=m`) to relay. By default, only hosts listed as RELAY in the access db file will be allowed to relay.

Relay based on MX Setting this option, will turn ON the ability to allow relaying based on the MX records of the host portion of an incoming recipient; that is, if an MX record for host foo.com points to your site, you will accept and relay mail addressed to foo.com.

Relay hosts only This option changes the behavior of the access database and class `R` macro to lookup individual host names only. By default, names that are listed as `RELAY` in the access database file and the class `R` (`$=R`) macro are domain names, and not host names.

Access db The access database (db) is a user-defined file to decide the domains from which you want to receive or reject mail messages. The entries in the access db file are either domain names, IP addresses, hosts names, or e-mail addresses. Every line of the access db file has a key and a value pair.

The key can be an IP address, a domain name, a hostname, or an e-mail address. The value part of the database can contain the following values:

`OK`	Accepts mail even if other rules in the running ruleset reject it. For example, if the domain name is unresolvable.
`RELAY`	Accepts mail addressed to the indicated domain or received from the indicated domain for relaying through your SMTP server. RELAY also serves as an implicit OK for the other checks.
`REJECT`	Rejects the sender or recipient with a general purpose message.
`DISCARD`	Discards the message completely using the `$#discard` mailer. This value works only for sender addresses (that is, it indicates that you must discard anything received from the indicated domain).
`### any text`	`###` specifies an RFC 821-compliant error code and `any text` specifies is a message to return for the command.

The default access db file is `/etc/mail/access`. You have to make a direct modification to `/etc/mail/sendmail.cf` if you want to use a non-standard access database filename.

NOTE

Because `/etc/mail/access` is a database, after creating the text file, you must use the following `makemap` command to create the database map.

```
makemap dbm /etc/mail/access < /etc/mail/access
```

For more information on the `makemap` utility, type `man 1M makemap` at the HP-UX prompt.

Relay local from This option allow Sendmail to relay mail messages when the sender of the mail message is a valid user on that machine. Consider a valid user `abc` on host 1. A user `cbz` on host 2 can connect to host 1 as user `abc` and send mail to another user `xyz` on host 3. This means that host 1 is now acting as a local relay agent.

You must enable this option only if absolutely necessary because it opens a window for spammers. Specifically, spammers can send mail to your mail server that claims to be from your domain (either directly or through a routed address), and you can then go ahead and relay it out to arbitrary hosts on the Internet.

Blacklist recipients This feature enables Sendmail to block incoming mail messages destined to certain recipient user names, host names, or addresses. This feature also restricts you from sending mail messages to addresses with an error message or REJECT value in the access database file. For example, if you have the following entries in the access database file:

```
badlocaluser        550 Mailbox disabled for this username
host.mydomain.com   550 That host does not accept mail
user@otherhost.mydomain.com   550 Mailbox disabled for this
recipient
```

These entries prevent a recipient of `badlocaluser@mydomain.com`, any user at `host.mydomain.com`, and the single address `user@otherhost.mydomain.com` from receiving mail.

```
spammer@aol.com          REJECT
cyberspammer.com           REJECT
```

The entries in the access db file indicate that Sendmail cannot send mail messages to `spammer@aol.com` or to the domain `cyberspammer.com`.

Accept unresolvable domains Setting this option, allows Sendmail to accept all those `MAIL FROM:` parameters that are not fully qualified, that is, if the host portion of the argument to `MAIL FROM:` command cannot be located in the host name service (for example, DNS).

Accept unqualified senders This option allows Sendmail to accept all those `MAIL FROM:` parameters where the mail address of the sender does not include a domain name. Normally, `MAIL FROM:` commands in the SMTP session are refused if the connection is a network and the sender address does not include a domain name.

Realtime Blackhole List Setting this option, turns ON the rejection of hosts found in the Realtime Blackhole List. The default list is maintained on the server `$def_rbl`. This option has now been deprecated.

Loose relay check This option turns off the default behavior of rechecking all those recipients using the % addressing. For example, if the recipient address is `user%site@othersite`, the default behavior without the loose_relay_check option is that Sendmail will check if any othersite is an allowed relay host specified in either class `R` macro or the access db file. If a site is an allowed relay host, the `check_rcpt` ruleset strips `@othersite` and checks `user@site` for relaying. Sendmail does not recheckif this option is set to ON. This option is not required for most installations.

Promiscuous Relay This option allows your mail server to relay any received mails. You must be careful before enabling this option.

No Default MSA You can use this option to generate the configuration file without the `DaemonPortOptions` option for the Message Submission Agent (MSA) daemon. If you use this option, the `sendmail.cf` configuration file will not contain the following line:

```
O DaemonPortOptions=Port=587, Name=MSA, M=E
```

DNS Blackhole List The `dnsbl` option avoids the possible confusion between `RealtimeBlackhole` List and other DNS-based Blacklist servers, such as ORBS. It takes the name of the Blacklist server and also an optional rejection message as arguments.

You can include `dnsbl` multiple times in the `sendmail.cf` file, thereby allowing sites to subscribe to multiple Blacklist servers. The Blacklist server verifies the IP address of the incoming connection and rejects all the `SMTP` commands if the address is blacklisted. An error message is also displayed.

Relay mail from You can use this option to facilitate relaying through a user machine. The sender name, which is listed as RELAY in the access map (tagged with `From:`), can be specified using this option. The domain portion of the mail sender is also checked when the optional argument `domain` is provided.

Delay checks This option delays the anti-spam checks by Sendmail until it issues the `SMTP RCPT` command. Mail from certain addresses that might have been blocked by other anti-spam checks are received. In these cases, deferred checks are not done.

By using `delay_checks`, the rulesets `check_mail` and `check_relay` are not called when a client connects or issues a `MAIL` command, respectively. Instead, those rulesets are called by the `check_rcpt` ruleset; they are

skipped if a sender has been authenticated using a **trusted** mechanism, for example, one that is defined via the list of `AuthMechanisms`. If `check_mail` returns an error, the `RCPT TO` command is rejected with that error. If it returns some other result starting with `$#`, then `check_relay` is skipped. If the sender address (or a part of it) is listed in the access map and it has a RHS of OK or RELAY, then `check_relay` is skipped.

Ldap Routing You can use this option to implement the LDAP-based email recipient routing. This provides a method for rerouting addresses with a domain portion in class {`LdapRoute`} either to a different mail host or to a different address.

For more information, see “LDAP-Based Routing” on page 71.

Milertable This option includes a "mailer table" which can be used to override routing for particular domains (which are not in local host names).

Genericstable If the genericstable is enabled and GENERICS_DOMAIN or GENERICS_DOMAIN_FILE is used, this feature will cause addresses to be searched in the map if their domain parts are subdomains of elements in class {G}. For more information, see “Creating Domain-Specific Aliasing Using Virtual Hosting” on page 68.

Virtusertable If the virtusertable is enabled and VIRTUSER_DOMAIN or VIRTUSER_DOMAIN_FILE is used, this feature will cause addresses to be searched in the map if their domain parts are subdomains of elements in class {VirtHost}. For more information, see “Creating Domain-Specific Aliasing Using Virtual Hosting” on page 68.

Domaintable Include a "domain table" which can be used to provide domain name mapping. Use of this should really be limited to your own domains. It may be useful if you change names (for example, your company changes names from oldname.com to newname.com).

Send only This option generates a `sendmail.cf` file without the `check_compat` ruleset. You can send mail messages, but you cannot receive them.

You must set the `SENDMAIL_SENDONLY` flag in `/etc/rc.config.d/mailservs` file to 1 in order to use the `send_only` feature.

Receive only This option generates a `sendmail.cf` file with a new set of rules called `check_compat`. You can receive mail messages, but you cannot send them. The following are added in the `/etc/rc.config.d/mailservs` file:

- `SENDMAIL_RECVONLY`

  You must set this flag to 1 in order to use the `receive_only` feature.

- `SENDMAIL_SENDONLY`

  You must set this flag to 1 in order to use the `send_only` feature.

NOTE

Sendmail depot installs the `mailservs` file in the directory `/usr/newconfig/etc/rc.config.d`. You must manually move this file to `/etc/rc.config.d/` in order to use this feature.

The priorities for these flags are defined in the `/usr/newconfig/etc/rc.conig.d/mailservs` file.

Creating Sendmail Aliases

The Sendmail aliases database stores mailing lists and mail aliases. You must create the aliases database by adding aliases to the file `/etc/mail/aliases` and then by running the `/usr/sbin/newaliases` command to generate the database from the file. The generated alias database is stored in the file `/etc/mail/aliases.db`. The Sendmail startup script also generates the aliases database when you reboot your system.

Each user on your system can create a list of alternate mailing addresses in a `.forward` file in the user's home directory. The `.forward` file allows users to forward their own mail to files or to other mailing addresses.

This section discusses the following topics:

NOTE A non-root user does not have access to the files or databases associated with Sendmail namely: `/etc/mail/aliases.*`, `/etc/mail/sendmail.st`, and `/etc/mail/sendmail`.

Adding Aliases to the Sendmail Alias Database

To add Sendmail aliases to the database, follow these steps:

1. If the file `/etc/mail/aliases` does not exist on your system, copy it from `/usr/newconfig/etc/mail/aliases` to `/etc/mail/aliases`.
2. Use a text editor to edit the file. Each line is of the following form:

   ```
   alias: mailing_list
   ```

where `alias` is the local address, local user name, or local alias, and `mailing_list` is a comma-separated list of local user names or aliases, remote addresses, file names, commands, or included files. Table 2-1 describes the options that can be included in a mailing list.

3. Issue the following command to regenerate the aliases database from the `/etc/mail/aliases` file:

   ```
   /usr/sbin/newaliases
   ```

   This command creates the aliases database located in `/etc/mail/aliases`.

Table 2-1 Mailing List Options

Option	Description

user_name	Sendmail looks up the aliases database for the local user name unless you put a backslash (\) before the local user name. To prevent Sendmail from performing unnecessary alias lookups, put backslashes before local user names. For example:
	`local_users: \amy, \carrie, \sandy, \anne,\david,\tony`
	`remote_users: mike, denise`
	`mike: mike@chem.tech.edu`
	`denise: bigvax!amlabs!denise`

remote_address	The remote address syntax that Sendmail understands is configured in the Sendmail configuration file and usually includes RFC 822 style addressing (*user*`@`*domain*) and UUCP style addressing (*host*`!`*user*). For example: `chess_club: mike@chem.tech.edu, marie@buffalo, bigvax!amlabs!denise`

Table 2-1 Mailing List Options (Continued)

Option	Description

filename	An absolute pathname on the local machine. Sendmail appends the message to the file if the following conditions are true:
	• The file exists, is not executable, and is writable by all.
	• The directory where the file resides is readable and searchable by all. Example:
	`public: /tmp/publicfile`
	`terminal: /dev/tty`
	Mail addressed to `public` is appended to `/tmp/publicfile`. Mail addressed to `terminal` appears on the sender's terminal.

"| *command*"	Sendmail pipes the message as standard input to the specified command. The double quotes are required to protect the command line from being interpreted by Sendmail. Commands must be listed as full pathnames. If `stdout` and `stderr` are not redirected, they are not printed to the terminal, and they disappear. However, if a command returns a nonzero exit status, its output to `stderr` becomes part of the Sendmail error transcript. The command is executed by the `prog` mailer defined in the configuration file. In the configuration file supplied with HP-UX, the `prog` mailer is configured as "`sh -c`". For example: `prog: "| / usr /bin/cat | / usr /bin/sed 's/Z/z/g' > /tmp/outputfile"` Mail addressed to `prog` is saved in `/tmp/outputfile` with all capital Z's changed to lowercase z's.

Table 2-1 Mailing List Options (Continued)

Option	Description

`:include:`*`filename`*	Any mail addressed to the alias is sent to all the recipients listed in the included file. The file must be a full pathname. Nonroot users can create `:include` files to maintain their mailing lists. An `:include` file can contain anything that is specified in the right side of an alias definition. Following is an example alias definition:
	`dogbreeders: :include:/users/andrea/dogbreeders`
	Following is an example `:include` file:
	`#file included in dogbreeders alias definition: terriers@akc.ny.com, coonhounders@ukc.sc.com`

An alias can be continued across multiple lines in the aliases file. Lines beginning with blanks or tabs are continuation lines.

The aliases file can contain comment lines, which begin with the pound sign (`#`). Blank lines in the aliases file are ignored.

NOTE You cannot address messages directly to file names, command lines, or `:include` files. Sendmail will deliver messages to these only if they appear in the right side of an alias definition.

Configuring Owners for Mailing Lists

Sendmail enables you to configure an owner for a mailing list, because the sender of a message often does not control the mailing list to which the message is addressed. If Sendmail encounters an error while attempting to deliver a message to the members of a mailing list, it looks for an alias of the form `owner-`*`mailing_list`* and sends the error message to the owner. For example, if `mike` were responsible for maintaining the `chess_club` mailing list, he could be configured as the owner:

```
chess_club:  mike@chem.tech.edu, marie@buffalo,
bigvax!amlabs!denise, margaret@hp.com
owner-chess_club:  mike@chem.tech.edu
```

Any errors that Sendmail encounters while trying to deliver mail to the members of the `chess_club` mailing list would be reported to `mike`.

Avoiding Alias Loops

You must avoid creating aliasing loops. Loops can occur either locally or remotely. An example of a local alias loop is as follows:

```
#Example of a local alias loop
first : second
second : first
```

While regenerating the alias database, the `newaliases` command does not notice a loop like the one shown in the previous example. However, after the alias database is generated, mail addressed to either `first` or `second` is not sent. If the recipients for the message are only in the local alias loops, the message is returned with the error message `All recipients suppressed`.

In the previous example, if mail is addressed to `first`, `first` expands to `second`, which expands back to `first`. This causes Sendmail to remove `first` from the recipient list as a duplicate.

```
# Example alias entry on host sage
dave : dave@basil
# Example alias entry on host basil
dave : dave@sage
```

The following is an example of a remote aliasing loop:

Mail sent to `dave` at either host `sage` or host `basil` bounces between the two systems. Sendmail adds a tracing header line (`Received:`) with each hop. When 26 tracing header lines have been added, Sendmail recognizes the aliasing loop and aborts the delivery with an error message.

Creating a Postmaster Alias

RFC 2822 requires that a postmaster alias be defined on every host. The **postmaster** is the person in charge of handling problems with the mail system on that host. The default aliases file supplied with the HP-UX operating system designates the postmaster as root. You can change this alias to the appropriate user for your system.

Verifying Your Sendmail Aliases

After you have created a Sendmail alias and regenerated the aliases database, issue the following command to verify the validity of your alias:

```
/usr/sbin/sendmail  -bv  -v  alias, alias, . . .
```

The `-bv` option causes Sendmail to verify the aliases without collecting or sending any messages. Any errors in the specified aliases are logged to standard output.

You can use the HP `expand_alias` utility to expand an alias or mailing list as far as possible. For more information on the `expand_alias` utility, type `man 1M expand_alias` at the HP-UX prompt.

Managing Sendmail Aliases with NIS or NIS+

You can manage the Sendmail aliases database through the Network Information Service (NIS or NIS+), which is one of the NFS Services. This service allows you to maintain an aliases database on one server system. All other systems request alias information from the server. In order to use NIS or NIS+, you must set up an NIS or NIS+ domain and configure the machines in your network as NIS or NIS+ servers and clients. For information about the NIS or NIS+ aliases database, see the manual *Installing and Administering NFS Services*, at the URL `http://www.docs.hp.com/hpux/onlinedocs/B1031-90048/B1031-90048.html`.

When you configure NIS or NIS+ on your network, it manages your Sendmail aliases by default, so you do not have to make any changes to your NIS or NIS+ configuration.

Before you run the NIS `ypinit` script or the NIS+ `nispopulate` script, ensure that the `/etc/mail/aliases` file on the NIS or NIS+ master server contains all the Sendmail aliases that you want to make globally available through NIS or NIS+.

The Sendmail program uses the Name Service Switch to determine where to look for Sendmail aliases.

Modifying your NIS Aliases Database

For information about the NIS or NIS+ aliases database, see *Installing and Administering NFS Services*, at the URL `http://www.docs.hp.com/hpux/onlinedocs/B1031-90048/B1031-90048.html`.

Rewriting the From Line on Outgoing Mail

HP provides a method that allows the `From` line on a mail message to be rewritten. This can be useful when a user's login name does not clearly identify the user to intended mail recipients. For example, mail sent by `bkelley (mailname)` can be changed to read as `Bob_Kelley (maildrop)`.

To rewrite `From` lines on an outgoing mail message, do the following:

1. Create the file `/etc/mail/userdb`, which contains two entries for each mail user. The entries must be in the following format:

   ```
   bkelley:mailname      Bob_Kelley
   Bob_Kelley:maildrop   bkelley
   ```

2. Build the `/etc/mail/userdb.db` file with the `makemap` routine:

   ```
   makemap btree /etc/mail/userdb.db < /etc/mail/userdb
   ```

3. Uncomment the following line in the `/etc/mail/sendmail.cf` file:

   ```
   UserDatabaseSpec=/etc/mail/userdb.db
   ```

4. Add the `i` flag to all the mailer definitions, to enable UDB sender rewriting. For example, change the mailer definition from

   ```
   Mlocal, P=/usr/bin/rmail, F=lsDFMAw5:/|@m,
   S=10/30, R=20/40, T=DNS/RFC822/X-Unix,
   A=rmail -d $u
   ```

   to

   ```
   Mlocal, P=/usr/bin/rmail, F=lsDFMAw5:/|@mi,
   S=10/30, R=20/40, T=DNS/RFC822/X-Unix,
   A=rmail -d $u
   ```

5. Uncomment the first rule in ruleset 94.

Forwarding Your Own Mail with a .forward File

You can redirect your own mail by creating a `.forward` file in your home directory. If a `.forward` file exists in your home directory and is owned by you, Sendmail redirects mail addressed to you to the addresses that the `.forward` file contains.

A `.forward` file can contain anything that appears on the right side of an alias definition, including programs and files. (See Table 2-1 earlier in this chapter.) The following is an example of a `.forward` file owned by user `alice` on host `chicago`:

```
alice@miami, alice@toronto, \alice,  mycrew
```

Mail sent to `alice@chicago` will be delivered to `alice`'s accounts on hosts `miami` and `toronto`, and to her account on local host `chicago`. It will also be delivered to all the recipients of the mailing list `mycrew`, which must be defined in the local aliases database or in the `:include` file on host `chicago`.

The aliases database is read before a `.forward` file. The `.forward` file is read only if the user's name is not defined as an alias or if an alias expands to the user's name.

Creating Domain-Specific Aliasing Using Virtual Hosting

Sendmail controls the `/etc/mail/virtusertable` database. This database provides a domain-specific form of aliasing and also allows multiple domains to be hosted on a single machine.

With this feature, users can have their own domain names and receive mail using these domain names with a single host. You are required to obtain a new (available) domain name and set up name servers for that domain. Then, you must configure MX records for your new domain.

NOTE Virtual hosting requires DNS to be set up. For information on setting up DNS, see the *IP Address and Client Management Administrator's Guide*, at the URL `http://www.docs.hp.com/hpux/netcom/index.html#Internet%20Services`

The following steps describe how to set up virtual hosting:

1. Assume `mydomain.com` as the new domain name. If the mail server that serves the new domain name has a full-time connection to the Internet, include the following line in the `db.domain` file (`domain` is the domain name specified in the file `/etc/resolve.conf`):

   ```
   mydomain.com. IN MX 10 mymailserver.mydomain.com.
   ```

   Otherwise, you must have another machine to queue mail for your domain. Include the following lines in the `db.domain file`:

   ```
   mydomain.com. IN MX 10 mymailserver.mydomain.com.

   mydomain.com. IN MX 20 othermailserver.otherdomain.com.
   ```

   Now you must set up Sendmail.

2. Generate the `sendmail.cf.gen` file using the `gen_cf` utility with the `virtusertable` option, and move this file to `/etc/mail/sendmail.cf`.

   For more information on `gen_cf`, read the section "Modifying the Default Sendmail Configuration File" on page 48.

3. Create the virtual user table in `the /etc/mail` directory. A sample virtual user table may look like the following:

   ```
   joe@mydomain.com          jschmoe
   jane@mydomain.com         jdoe@othercompany.com
   @mydomain.com             jschmoe
   ```

   In this example, the address `joe@mydomain.com` is mapped to the local user `jschmoe`, `jane@mydomain.com` to the remote user `jdoe@othercompany.com`, and any other address in `mydomain.com` is mapped to `jschmoe`.

4. Build the virtual user table database file by running the `makemap` utility on the command line as follows:

   ```
   # makemap dbm /etc/mail/virtusertable < /etc/mail/virtuser
   table
   ```

   To reverse map local users for outbound mails, you must generate the `sendmail.cf` file with the `genericstable` option in addition to the `virtusertable` option.

   You must generate the generics table similar to the virtual user table, but with the entries reversed.

   Example:

   ```
   jschmoe                   joe@yourdomain.com
   ```

5. Add your domain name to the `/etc/mail/sendmail.cw` file.

6. Kill and restart Sendmail.

   You can now receive mail at `mydomain.com`.

IMPORTANT

The virtual hosting feature provides better support for ISPs that offer queuing services to dial-up customers because queue-runs no longer wait for the dial-up server connection attempts to time out.

Sendmail and the LDAP Protocol

The Lightweight Directory Access Protocol (LDAP) enables servers to share static information. Combining Sendmail and LDAP increases the speed and efficiency at which network information is collected and displayed.

Sendmail supports the use of the LDAP protocol to look up addresses. The `ldapx` class, which is a database, is used to look up items in the LDAP directory service. The `Sendmail` configuration file contains the syntax required to enable the LDAP protocol to perform address lookups.

Enabling Address Lookups Using LDAP

When you enable LDAP support, LDAP will look up login names, then return the e-mail address for that user. To enable this, you must modify the `sendmail.cf` file.

The following steps describe how to enable address lookup using LDAP:

1. Open the `sendmail.cf` file.
2. Uncomment the following ruleset:

   ```
   #R$+ < @ $+ > $: $: $(ldap $1 $:  $1<@$2>$) ldap support
   ```

3. Uncomment the following line:

   ```
   Kldap dapx -k"uid=%s" -v"mail" -htest.india.hp.com" -b"org
   anization, c=US"
   ```

This enables the LDAP protocol to perform lookups. These lookups are defined entirely by the switches specified. In the previous example, `-k` and `-v` are the switch options.

The `-k` switch defines how the map takes its input value and constructs the LDAP search. The `-v` switch is the value that replaces the original string in the map. In most cases, this will be an e-mail address. The `-b` switch is the directory in the LDAP tree where searching begins. The `-h` switch is the space-separated string of servers that support LDAP at your site.

NOTE The LDAP-style options (`-v` and `-h` in the previous example) must be double quoted and must follow immediately after the option. Do not leave spaces between the option and the quote.

LDAP-Based Routing

You can use the LDAP protocol to implement LDAP-based rerouting. This provides a method to reroute addresses with a domain portion in class `{LDAPRoute}` to either a different mail host or a different address.

You can use the `/usr/newconfig/etc/mail/cf/cf/gen_cf` script to enable the LDAP-based routing.

You can add the domains to the class `{LDAPRoute}`, as shown in the following examples. Ensure that you set up a domain for LDAP routing. Assume that your domain is `yyy.com`. Add the following line in the `sendmail.cf` file:

```
C{LDAPRoute}yyy.com
```

or

```
F{LDAPRoute}/etc/mail/ldap-domain-file
```

where `/etc/mail/ldap-domain`-file contains the domains.

The `LDAPDefaultSpec` option in the `sendmail.cf` file sets the default LDAP map specification. You must set this up before defining LDAP maps. The settings are used for all LDAP maps unless they are specified in the individual map specification (`K` command). By default, it appears in the `sendmail.cf` file as follows:

```
O LDAPDefaultSpec=-h localhost
```

`localhost` can be replaced by your LDAP server name.

Following are the switches commonly used by most applications:

- `-b` – LDAP search base

  Directory in the LDAP tree where the search begins. For example:

  ```
  -b "o=hp.com"
  ```

- `-d` – BindDN

The BindDN parameter used to specify the DN value for the LDAP bind request. For example:

```
-d"cn=ldap://:389,dc=edat104,dc=atl,dc=hp,dc=com"
```

- `-h` – LDAP servers

  Space-separated string of servers that support LDAP at your site. For example:

  ```
  -h "ldap1.hp.com ldap2.hp.com"
  ```

- `-p` – Port numbers

  Port numbers where LDAP service is available. For example:

  ```
  -p 33333
  ```

- `-k` – LDAP search string (key)

  String that defines how an LDAP map takes its input value and initiates an LDAP search. For example:

  ```
  -k (&(ObjectClass=mailrecipient) (mail=%0))
  ```

- `-v` – LDAP attribute

  Value that replaces the origin string in the map. In most cases, this is the RFC822 e-mail address. For example:

  ```
  -v mailroutingaddress
  ```

The LDAP maps are defined in the configuration file as follows:

```
Kldap -1 -v mailHost -k (&(objectClass=inetLocalMailRecipient)
 (mailLocalAddress=%0))
```

```
Kldapmra ldap -1 -v mailRoutingAddress -k (&(objectClass=inetL
ocalMailRecipient) (mailLocalAddress=%0))
```

`mailLocalAddress` is the RFC 2822-compliant e-mail address of the recipient.

`mailHost` is the fully qualified host name of the MTA that is the final SMTP destination of the message to the recipient.

`mailRoutingAddress` is the RFC 822 address to be used when routing messages to the SMTP MTA of the recipient.

IPv6 Support

An option value `inet6` is provided for the field `Family` in `DaemonPortOptions` to enable IPv6 functionality.

To enable IPv6, set the `DaemonPortOptions` in the `sendmail.cf` configuration file as follows:

```
O DaemonPortOptions=Port=smtp, Name=MTA, Family=inet6
```

This will enable Sendmail to accept both IPv4 and IPv6 addresses.

Security

This section discusses administering Sendmail security options. It discusses the following topics:

Using the Sendmail Restricted Shell Program

Sendmail allows the `aliases` file or a user's `.forward` file to specify programs to be run. These programs are by default invoked through `/usr/bin/sh -c`. The Sendmail restricted shell (`smrsh`) program enables you to restrict the programs that can be run through the `aliases` file or through a `.forward` file; only programs that are linked to the `/var/adm/sm.bin` directory can be invoked.

To use the `smrsh` program, complete the following steps:

1. In the `/etc/mail/sendmail.cf` file, comment the following lines by inserting a pound sign (#) before each line:

   ```
   # Mprog, P=/usr/bin/sh, F=lsDFMoeu, S=10/30, R=20/40, D=$z
   :/,

   # T=X-Unix,

   # A=sh -c $u
   ```

2. In the `/etc/mail/sendmail.cf` file, uncomment the following lines by deleting the pound sign (#) before each line:

   ```
   Mprog, P=/usr/bin/smrsh, F=lsDFMoeu, S=10/30, R=20/40, D=$
   z:/,

   T=X-Unix,

   A=smrsh -c $u
   ```

3. Create the directory `/var/adm/sm.bin/` with root:bin ownership and 755 permissions. Place the binaries of the programs that you want to allow into this directory. Typically, programs such as `vacation`,

rmail, and AutoReply are placed in this directory. (You can also specify hard links to the binaries.) Do not place shells such as ksh, sh, csh, and perl in this directory because they have too many security issues.

Turning Off Standard Security Checks

Sendmail has security checks that limit reading and writing to certain files in a directory. These checks protect files that may reside in unsafe directories or that may be tampered with by users other than the owner. You can turn these safety checks off by editing the DontBlameSendmail option in the configuration file.

In the sendmail.cf file, change DontBlameSendmail=*option value*, where *option value* is any of the options listed in Table 2-2. The default option value is *safe*. After you change *option value*, the new value becomes the default value.

Table 2-2 Option Values for DontBlameSendmail

Option Value	Description
safe	Allows the files only in a safe directory. All files accessed by Sendmail must be safe.
AssumeSafeChown	Assumes that the chown system call is restricted to root.
ClassFileInUnsafeDirPath	Allows class files that are in unsafe directories.
ErrorHeaderInUnsafeDirPath	Allows the file named in the ErrorHeader option to be in an unsafe directory.
ForwardFileInGroupWritableDirPath	Allows .forward files in group-writable directories.
GroupWrtableDirPathSafe	Considers group-writable directories to be safe. Sendmail will read messages from group-writable directories.
GroupWritableIncludeFileSafe	Accepts group-writable :include files
GroupWritableAliasFile	Allows group-writable alias files.

Table 2-2 Option Values for DontBlameSendmail (Continued)

Option Value	Description
`HelpFileinUnsafeDirPath`	Allow Help file to be in unsafe directory.
`IncludeFileInGroupWritableDirPath`	Allows `:include:` files in group-writable directories.
`ForwardFileInUnsafeDirPath`	Allows a `.forward` file that is in an unsafe directory to include references to programs and files.
`IncludeFileInUnsafedirPathSafe`	Allows an `:include:` file that is in an unsafe directory to include references to programs and files.
`MapInUnsafeDirPath`	Allows maps (for example, hash, btree, and dbm files) in unsafe directories.
`LinkedAliasFileInWritableDir`	Allows an alias file that is a link in a writable directory.
`LinkedClassFileInWritableDir`	Allows class files that are links in writable directories.
`LinkedForwardFileInWritableDir`	Allows `.forward` files that are links in writable directories.
`LinkedIncludeFileInWritableDir`	Allows `:include:` files that are links.
`LinkedMapInWritableDir`	Allows map files that are links in writable directories.
`LinkedServiceSwitchFileInWritableDir`	Allows the service switch file to be a link even if the directory is writable.
`FileDeliveryToHardLink`	Allows delivery to files that are hard links.
`FileDeliveryToSymLink`	Allows delivery to files that are symbolic links.
`WriteMapToHardLink`	Allows writes to maps that are hard links.
`WriteMapToSymLink`	Allows writes to maps that are symbolic links.

Table 2-2 Option Values for DontBlameSendmail (Continued)

Option Value	Description
`WriteStatsToHardLink`	Allows the status file to be a hard link.
`WritesStatsToSymLink`	Allows the status file to be a symbolic link.
`RunProgramInUnsafeDirPath`	Allows Sendmail to run programs that are in writable directories.
`RunWritableProgram`	Allows Sendmail to run programs that are group- or world-writable.
`WorldWritableAliasFile`	Accept world-writable alias files.

Disabling Privacy Options

You can now disable the ETRN and VERB privacy options by using the `noetrn` and `noverb` flags:

- `PrivacyOptions=noetrn`

  The `noetrn` flag disables the SMTP ETRN command, enabling Sendmail to process its queue in a synchronous mode.

- `PrivacyOptions=noverb`

  The `noverb` flag disables the SMTP VERB command, turning off verbose mode.

For more information on the different privacy options, see the Sendmail configuration file `/etc/mail/sendmail.cf`.

Enabling SMTP Authentication Based on RFC 2554

A new option to set AUTH parameter in `MAIL FROM` command has been added in the `sendmail.cf` file. By default, this appears as follows:

```
#O AuthOptions
```

Sendmail supports `SMTP AUTH` as defined in **RFC 2554** (*SMTP Service Extension for Authentication*), which is based on *Simple Authentication and Security Layer* – **RFC 2222** (SASL). SMTP authentication provides a robust tool to control relaying with maximum flexibility. SASL is

mainly used for roaming users whose IP address and host name changes repeatedly. In this case, authorization is via a secret password, which is client dependent.

The authentication protocol exchange consists of a series of server challenges (otherwise known as a ready response) and client answers that are specific to the authentication mechanism.

The AUTH parameter to the MAIL FROM command is set as follows:

```
MAIL FROM: from-addr AUTH=addr-spec
```

The addr-spec contains the identity that submitted the message to the delivery system. If the server trusts the authenticated identity of the client to assert that the message was originally submitted by the supplied addr-spec, then the server must supply the same addr-spec in an AUTH parameter when relaying the message to any server that supports the AUTH extension.

You can specify the list of authentication mechanisms for AUTH in the AuthMechanisms option in the sendmail.cf file. By default, it appears in the sendmail.cf file as follows:

```
#O AuthMechanisms=GSSAPI KERBEROS_V4 DIGEST-MD5 CRAM-MD5
```

If you set this option to A, the AUTH= parameter for the MAIL FROM command is issued only when authentication succeeds.

DaemonPortOptions has a suboption called modifiers (M). The modifiers suboption contains an authentication flag a, which instructs the daemon to authenticate all its connections.

By default, it appears in the sendmail.cf file as:

```
#O DefaultAuthInfo=/etc/mail/default-auth-info
```

The DefaultAuthInfo option sets the file name, which by default contains the authentication information for outgoing connections. It must contain the authorization ID (userid), the authentication ID (authid), the password (plain text), and the realm to use, each on a separate line. This information must be readable only by root (or by the trusted user). If you do not specify a realm, $j is used.

Support for RFC 1413 (Identification Protocol)

`identd` is a server that implements the TCP/IP proposed standard IDENT user identification protocol as specified in **RFC 1413**. `identd` listens on port 113 and operates by looking up specific TCP/IP connections and returning the user owing the process owning the connection.

Sendmail uses `identd` as an advisory mechanism to log the identity of the user name and host name of the Sendmail client. `identd` may cause additional traffic for collecting the user name, which may adversely affect the performance of Sendmail.

Enabling identd on the Sendmail Server

You can enable `identd` on the Sendmail server by uncommenting the following entry in the `/etc/mail/sendmail.cf` file:

```
#O Timeout.ident=5s
```

By default, the `identd` timeout value is 5 seconds.

You can disable `identd` to improve the performance of the system by commenting out this entry. The following sections discuss disabling `identd`:

- "Disabling identd on the Remote Client" on page 79
- "Disabling identd from the Sendmail Server" on page 80

Disabling identd on the Remote Client

You must comment out the following line in the `/etc/inetd.conf` file in the client system, by placing a pound sign (#) in the first column as follows:

```
#auth stream tcp wait bin /usr/lbin/identd identd
```

The previous command denotes an IPv4 enabled system. If the system is IPv6 enabled, then you must comment out the following line:

```
#auth stream tcp6 wait bin /usr/lbin/identd identd
```

Then, execute the command `inetd -c` to restart the `inetd` daemon in the client system, thereby forcing `inetd` to reread the `inetd.conf` file.

Disabling identd from the Sendmail Server

This is probably an easier way of disabling `identd`, because you need not be concerned about the remote client having `identd` disabled. In the file `/etc/mail/sendmail.cf` on the Sendmail server, modify the following entry:

```
#O Timeout.ident=5s
```

as

```
O Timeout.ident=0s
```

Now, you need to kill and restart Sendmail.

Configuring Sendmail to Reject Unsolicited Mail

You can set up Sendmail so that unsolicited or **spam** mail (mail sent to large number of users) is not transmitted to or received by users on the network.

The first step in configuration is to enable the anti-spamming rulesets. You then edit other configuration files to control mail transmission. This section describes how to:

- Accept or reject mail from particular senders
- Prevent your machine from being used as a relay machine
- Accept or reject connections from specific users' host names based on domains or IP addresses
- Enable or disable mail transfers from specific senders and recipient pairs

The anti-spamming features enable you to control the users who can send, receive, or relay mail messages on the network. This section discusses the following topics:

- "Enabling Anti-Spamming Security Features" on page 81
- "Using the Access Database to Allow or Reject Mail Messages" on page 82
- "Enabling Anti-Spamming Relay Features" on page 85
- "Validating Senders" on page 86
- "Checking Headers" on page 88
- "Spam Control Using the Message Submission Agent (RFC 2476)" on page 90

Enabling Anti-Spamming Security Features

You must run the `gen_cf` script to turn on relaying, validating, and checking features.

The access database also allows you to control the message flow. See the section "Using the Access Database to Allow or Reject Mail Messages" on page 82 for more information.

Running the gen_cf Script

Follow these steps to run the `gen_cf` script:

1. Log in as `root`.
2. Go to the directory that contains the script:

   ```
   cd /usr/newconfig/etc/mail/cf/cf/gen_cf
   ```

3. Run `gen_cf`.
4. A list of options is displayed. Select the appropriate option.

A message is displayed to inform you when the file is successfully built.

Using the Access Database to Allow or Reject Mail Messages

You can control the flow of mail messages coming in from certain domains. The Access Database enables you to allow or reject mail from specific domains. By default, names listed in the database as OK are domain names, not host names.

Following are the steps to allow or reject messages:

1. Create an access database text file.
2. Create a database map.

You must understand a few basic facts about the Access Database format and structure before creating the Access Database file or database map.

Access Database Format

This section includes a few key points about the database and describes the format of the database.

- Every line of the access database file has a key and a value pair.
- The value part of the database can be any of the values listed in Table 2-3.

The key can be an IP address, a domain name, a host name or an e-mail address.

Table 2-3 Access Database Format

Value	Description
`OK`	Accepts mail even if other rulesets rejects it. For example, if the domain name is unresolvable.
`RELAY`	Accepts mail addressed to the specified domain or received from the specified domain for relaying through your SMTP server. RELAY also serves as an implicit OK for the other checks.
`REJECT`	Rejects the sender or recipient with a general-purpose message.
`DISCARD`	Discards the message completely using the $#discard mailer delivery agent. This only works for sender addresses. That is, it indicates that you must discard anything received from the specified domain.
`### "any text"`	Where `###` is an RFC 821-compliant error code and "`any text`" is a message to return for the command.
ERROR: ### "any text"	Same as stated for `### "any text"`, but useful to mark error messages
ERROR:D.S.N:### "any text"	Same as stated for `### "any text"`. D.S.N is an RFC 1893-compliant error code.

Creating the Access Database Text File

You must edit the Access Database text file manually. The default Access Database file is `/etc/mail/access`. However, you can specify another file in the `sendmail.cf` file.

Table 2-4 contains a sample access database file, `/etc/mail/access`.

Table 2-4 Access Database Text File Example

`cyberspammer.com`	`550 We don't accept mail from spammers`
`okay.cyberspammer.com`	`OK`
128.32	`RELAY`
`spammer@aol.com`	`REJECT`
192.168.212	`DISCARD`

In the example Access Database text file, all mail messages from the `cyberspammer.com` domain are rejected and the error message `550 We don't accept mail from spammers` is displayed. All mail messages from the `okay.cyberspammer.com` domain are accepted. Messages can be relayed through 128.32. All mail messages from `spammer@aol.com` are rejected. All mail messages from the 192.168.212 domain are discarded.

Creating Finer Spam Control Using Tags

You can also tag entries in the access map based on their type. The following tags are available:

- `Connect:` connection information (`${client_addr}`, `${client_name}`)
- `From:` sender
- `To:` recipient

When the required item is looked up in a map, it is tried with the corresponding tag in front, then without any tag (as fallback to enable backward compatibility). For example:

```
From:spammer@some.dom  REJECT
To:friend.domain      RELAY
Connect:friend.domain  OK
Connect.from.domain    RELAY
From:good@another.dom   OK
From:another.dom     REJECT
```

Creating the Database Map

After creating the Access Database text file, you must use the `/usr/sbin/makemap` utility to create the database map. Type the following command to create the database:

```
makemap dbm /etc/mail/access < /etc/mail/access
```

The `makemap` utility takes `/etc/mail/access` file as input. It then stores the results back into the `/etc/mail/access.db` file.

Enabling Anti-Spamming Relay Features

The `gen_cf` shell script distributed with Sendmail enables you to turn on one or more of the following anti-spamming relay features:

- Promiscuous Relay: Relaying from Any Host to Any Host
- Relay Entire Domain: Relaying from Any Host in the Domain
- Relay Hosts Only: Relaying from Hosts Only
- Relaying Based on MX Records
- Relay from Local
- Check Loose Relay

Promiscuous Relay: Relaying from Any Host to Any Host

Promiscuous relay allows you to configure your site to allow mail relaying from any one site to any other site. This feature is not enabled by default.

You can enable promiscuous relay by choosing it as an option when running the `gen_cf` script distributed with Sendmail. When you enable this option, Sendmail does not check for relaying. Spammers may then relay mail through your site.

Relay Entire Domain: Relaying from Any Host in the Domain

By default, only hosts listed as RELAY in the Access Database are allowed to relay messages. The hosts must be defined in the `m` class (`$=m`) macro to relay. However, this feature allows any host in your domain to relay mail messages.

Relay Hosts Only: Relaying from Hosts Only

By default, host names that are listed as RELAY in both the Access Database and the class `R($=R)` macro can relay messages. When using this feature, specify host names. This feature enables Sendmail to look up individual host names and relay messages to the host.

See "Checking Headers" on page 88 for information on using the `R` class.

Relaying Based on MX Records

This feature allows relaying based on the MX records of the host portion of an incoming recipient. If an MX record for host `foo.com` points to your site, you will accept and relay mail addressed to `foo.com`.

Relay from Local

With this feature, a sender who is a valid user on a particular host can relay messages to other users on different hosts.

IMPORTANT

Use caution when using this feature. Using this feature opens a window for spammers. Specifically, spammers can send mail to your mail server that claim to be from your domain (either directly or via a routed address), and your machine will relay it out to any hosts on the Internet.

Check Loose Relay

This feature turns off the default behavior, which rechecks all recipients using `%` addressing. For example, if the recipient address is `user%site@othersite`, and *`othersite`* is in class `R` macro, Sendmail strips the `@othersite` portion and rechecks `user@site` for relaying.

Validating Senders

Sendmail provides a stringent check of mail message senders to ensure that they are legitimate. Sendmail refuses mail if the MAIL FROM: parameter has an unresolvable domain. You can work around this. If you want to continue accepting mail from such domains, use the features described in this section. You can enable any of the following features when you run the `gen_cf` script:

- Accept Unresolvable Domains

- Accept Unqualified Senders
- Blacklist Recipients
- Realtime Blackhole List

Accept Unresolvable Domains

This feature enables Sendmail to accept all `MAIL FROM:` parameters that are not fully qualified, for example, a mail message whose host part of the argument to the `MAIL FROM:` parameter cannot be located in the host name service, such as DNS.

Accept Unqualified Senders

This feature allows you to accept all mail where the sender's mail address does not include a domain name.

Normally, the `MAIL FROM:` commands in the SMTP session are refused if the connection is a network connection and the sender address does not include a domain name.

Blacklist Recipients

This feature enables Sendmail to block incoming mail messages destined for certain recipient user names, host names, or addresses. This feature also restricts you from sending mail messages to addresses with an error message or REJECT value in the Access Database file.

Example 1

For example, given the following entries in the Access Database file:

`badlocaluser`	`550 Mailbox disabled for this username`
`host.mydomain.com`	`550 That host does not accept mail`
`user@otherhost.mydomain.com`	`550 Mailbox disabled for this recipient`

Recipient of `badlocaluser@mydomain.com`, any user at `host.mydomain.com`, and the single address `user@otherhost.mydomain.com` will not receive mail.

Example 2

```
spammer@aol.com        REJECT
cyberspammer.com         REJECT
```

Mail cannot be sent to `spammer@aol.com` or to anyone at `cyberspammer.com`.

Realtime Blackhole List

This feature rejects hosts listed in the Realtime Blackhole List, which is found in the Realtime Blackhole List server. The server is `blackholes.mail-abuse.org`.

To use this feature, you must add the following line to the DNS database:

```
1.5.5.192.blackholes.mail-abuse.org IN A 127.0.0.2
```

You can specify the Realtime Blackhole List servers in the `sendmail.cf` file.

Checking Headers

With header checking, you can reject mail messages based on the contents of their mail headers. Sendmail provides the syntax for limited header syntax checking. A configuration line of the form: `HHeader: $>Ruleset` causes the specified ruleset to be invoked on the header when read. Following is an example of header checking:

```
Validity of a Message-ID: header
#LOCAL_RULESETS
HMessage-Id: $>CheckMessageId
SCheckMessageId
R< $+ @ $+ >            $@ OK
R$*                     $#error $: 553 Header Error
```

If the previous lines are included in the `sendmail.cf` file, then all header messages of the form `Message-Id:` will call the ruleset `SCheckMessageID`, which checks for the validity of the `Message-Id` header.

Discard Mailer

Sendmail has defined a special internal delivery agent called `discard`. You can use this agent with the header-checking ruleset and `check` rulesets: `check_mail`, `check_rcpt`, `check_relay`, or `check_compat`.

If any of the `check` rulesets (`check_mail`, `check_rcpt`, `check_relay`, or `check_compat`) or the header-checking ruleset resolves a mail address to the `$#discard` mailer, then all the SMTP commands are accepted, but

the message is discarded. If only one of message recipients address resolves to the `$#discard` mailer, none of the recipients will receive the mail message.

Regular Expressions

You can use regular expressions with the new map class `regex`. Use the `regex` map to see if an address matches a certain regular expression. By using such a map in a `check` rulesets (`check_mail`, `check_rcpt`, `check_relay`, or `check_compat`), you can block a certain range of addresses that would otherwise be considered valid.

For example, if you want to block all senders with all numeric user names, such as `2312343@bigisp.com`, you would use `SLocal_check_mail` and the new `regex` map:

```
#LOCAL_CONFIG
 Kallnumbers regex -a@MATCH ^[0-9]+$
 LOCAL_RULESETS
 SLocal_check_mail          # check address against\
                             various regex checks
 R$*                        $:  $>Parse0 $>3 $1
 R$+ < @ bigisp.com.  >48   $:  $(allnumbers $1 $)
 R@MATCH                    $#error $:553 Header Error
```

Defining Hosts Allowed to Relay: Class R

You can use the `$=R` macro to define the hosts that are allowed to relay. The default file Sendmail uses to read values for the `$=R` macro is `/etc/mail/relay-domains`.

Queue Changes

This section describes miscellaneous enhancements to the queue option:

- The queue option allows multiple `-qI`, `-qR`, or `-qS` queue run limiters.
  For example, using `Sendmail -qRfoo -qRbar` will deliver mail to recipients with `foo` or `bar` in their address.
- The map flag `-Tx` appends `x` to lookups that return temporary failure. This is similar to the `-ax` flag, which appends `x` to lookups that return success.
- The `QueueSortOrder` option is case sensitive.

Spam Control Using the Message Submission Agent (RFC 2476)

Sendmail supports **RFC 2476**, a protocol for message submission. The anti-spam rulesets have been enhanced to improve the anti-spam capabilities. The RFC proposes a new standard for the Message Submission Agent (MSA). This is designed to replace the more general-purpose Mail Transfer Agent (MTA) as the first service to which a Mail User Agent (MUA) connects to deliver a mail message. The RFC also describes how the usual protocols for SMTP service must be tightened up at the point where mail enters the system, rather than being routed from one site to another. Sendmail also serves as a powerful tool to authenticate and control mail messages.

By default, MSA is defined in the `sendmail.cf` file as:

```
O DaemonPortOptions=Name=MSA, Port=587, M=E
```

where `Port 587` is reserved for e-mail message submission.

An MSA still uses the same rulesets for processing the message (and therefore still allows message rejection via the `check` rulesets). In accordance with the RFC, the MSA ensures that all domains in the envelope are fully qualified if the message is relayed to another MTA. It also enforces the normal address syntax rules and log error messages. In addition, you can request authentication before the messages are accepted by MSA by using the `M=a` modifier in the `DaemonPortOptions`.

NOTE

You can turn off MSA in the `sendmail.cf` file using the option, `no_default_msa` in the `gen_cf` script. For more information, see the `no_default_msa` option in “Modifying the Default Sendmail Configuration File” on page 48.

The `XUSR SMTP` command and the `-U` (initial user submission) command-line option are deprecated. Mail user agents must use the MSA (Message Submission Agent) for initial user message submission. XUSR may be removed in future releases. The next release of Sendmail will assume that any message submitted from the command line is an initial user submission and act accordingly.

Sendmail Validation

The `check_compat` ruleset compares all sender and receiver pairs before mail is delivered. It validates the mail based on the results of the comparison. It checks to see if host A can legally send a message to host B. `check_compat` is called for all mail deliveries, not just SMTP transactions.

`check_compat` is used in the following situations:

- A set of users who are restricted from sending mail messages to external domains need to send mail messages to internal domains. Both the sender and recipient addresses are checked to ensure that they are in the local domain.
- A particular user needs to ensure that he or she does not receive mail messages from a specific source.
- A particular host needs to ensure that external senders do not use that host as a a mail relay. The mail messages are screened based on the sender's host name.

Turning Off Virtual Interfaces

You can disable the ability to include all the interface names in the `$=w` macro on startup. Turning off virtual interfaces speeds up the startup process. However, if you turn virtual interfaces off, mail sent to those addresses will bounce back to the sender.

To turn off virtual interfaces, do the following:

1. Open the `sendmail.cf` file.
2. Uncomment the line `DontProbeInterfaces`.

By default, virtual interfaces are included in the `$=w` macro, which is defined in the `sendmail.cf` file. Sendmail searches for them during startup.

The host name is added to class `w` for the names of all interfaces unless the `DontProbeInterfaces` option is set. This is useful for sending mail to hosts, which have dynamically assigned names.

Troubleshooting Sendmail

This section describes the following techniques for troubleshooting `Sendmail`:

You must log in as superuser to perform all Sendmail troubleshooting.

Keeping the Aliases Database Up to Date

You must rebuild the aliases database if you have made changes to the aliases text file.

You must restart Sendmail after you change the configuration file or the aliases database.

Issue the following commands, on a standalone system or on the mail server, to rebuild the aliases database and restart Sendmail:

```
/sbin/init.d/sendmail stop
/sbin/init.d/sendmail start
```

Updating your NIS or NIS+ Aliases Database

If you are using NIS or NIS+ to manage your aliases database, see *Installing and Administering NFS Services*, at the URL `http://www.docs.hp.com/hpux/onlinedocs/B1031-90048/B1031-90048.html`.

Verifying Address Resolution and Aliasing

In order to deliver a message, Sendmail must first resolve the recipient addresses appropriately. To determine how Sendmail would route mail to a particular address, issue the following command:

```
/usr/sbin/sendmail  -bv  -v  -oL10  address [address...]
```

The `-bv` (verify mode) option causes Sendmail to verify addresses without collecting or sending a message.

The `-v` (verbose) flag causes Sendmail to report alias expansion and duplicate suppression.

The `-oL10` (log level) option sets the log level to 10. At log level 10 and above, `sendmail -bv` reports the mailer and host to which it resolves recipient addresses.

For hosts that resolve to IPC mailers, `MX` hosts are not reported when using verify mode, because `MX` records are not collected until delivery is actually attempted.

If the address is not being resolved as you expect, you may have to modify one or more of the following:

- The Sendmail configuration file
- The files or programs from which file classes are generated
- The name server configuration
- The UUCP configuration

More detailed information about how the configuration file is rewriting the recipient addresses is provided by address test mode:

```
/usr/sbin/sendmail  -bt
```

Verifying Message Delivery

You can observe Sendmail's interaction with the delivery agents by delivering the message in verbose mode, as in the following example:

```
/usr/sbin/sendmail -v  myname@hp.com
```

Sendmail is now ready for you to type a message. After the message, type a period (.) on an empty line to denote the end of the message, as in the following example:

```
This is only a test.
.
```

Sendmail responds with the following information:

```
myname@baby.com... Connecting to sys1.hp.com via esmtp...
220 sys1.baby.com ESMTP Sendmail 8.8.6 (PHNE_12345)/8.8.6 SMKi
t7.02; Wed, 23 Oct 2002 18:44:21 +0530 (IST)
250-sys1.baby.com Hello root@inet.baby.com [15.70.178.1940, pl
eased to meet you
>>MAIL From:<root@inet.baby.com> SIZE=21
250 <root@inet.baby.com>... Sender ok
>> RCPT To:<myname@baby.com>
250 <myname@baby.com>
>>DATA
354 Enter mail, end with "." on a line by itself
>>>.
250 SAA24294 Message accepted for delivery
myname@baby.com... Sent (SAA24294 Message accepted for deliver
y)
Closing connection to sys1.baby.com
QUIT
221 sys1.baby.com closing connection.
```

Sendmail has interfaces to three types of delivery agents. In verbose mode, Sendmail reports its interactions with them as follows:

- Mailers that use SMTP to a remote host over a TCP/IP connection (IPC mailers).

  In verbose mode, Sendmail reports the name of the mailer used, each `MX` host (if any) to which it tries to connect, and each Internet address it tries for each host. When a connection succeeds, the SMTP transaction is reported in detail.

- Mailers that run SMTP (locally) over pipes.

  The name of the mailer used and the command line passed to `exec()` are reported. Then the SMTP transaction is reported in detail. If the mailer returns an abnormal error status, that is also reported.

- Mailers that expect envelope information from the Sendmail command line and expect message headers and message body from standard input.

  The name of the mailer used and the command line passed to `exec()` are reported. If the mailer returns an abnormal error status, that is also reported.

Contacting the Sendmail Daemon to Verify Connectivity

It is possible to contact the Sendmail daemon and other SMTP servers directly with the following command:

```
telnet  host  25
```

Use this to determine whether an SMTP server is running on *host*. If not, your connection attempt will return the message `Connection refused`.

After you establish a connection to the Sendmail daemon, you can use the SMTP `vrfy` command to determine whether the server can route to a particular address. For example:

```
telnet furschlugginer 25
220 furschlugginer.bftxp.edu ESMTP Sendmail 8.11.1/8.11.1; Wed
, 28 Aug 2002 14:33:50 +0530 (IST)

vrfy istm@hp.com
250 2.1.5 <istm@hp.com>

vrfy blemph@morb.poot
554 5.1.1 blemph@morb.poot... User unknown

quit
221 2.0.0 furschlugginer.bftxp.edu closing connection
Connection closed by foreign host
```

Not all SMTP servers support the `VRFY` and `EXPN` commands.

Setting Your Domain Name

If Sendmail cannot resolve your domain name, you may see the following warning message in your `syslog` file:

```
WARNING: local host name name is not qualified; fix $j in  con
fig file
```

To resolve this problem, do one of the following:

- Uncomment the following line in the `/etc/mail/sendmail.cf` file by deleting the pound sign (#) at the beginning of the following line:

  ```
  Dj$w.Foo.COM
  ```

  Change `Foo.COM` to the name of your domain (for example, `HP.COM`).

- Modify the `/etc/hosts` file, making sure that the fully qualified name of the system is listed first. For example, the entry in the file must be `255.255.255.255 dog.hp.com dog` and not `255.255.255.255 dog dog.hp.com`.

Attempting to Start Multiple Sendmail Daemons

If you attempt to invoke Sendmail when a Sendmail daemon is already running, the following message may be logged to the syslog file:

```
NO QUEUE: SYSERR (root) opendaemonsocket: daemon MTA: server S
MTP socket wedged: exiting
```

This message means that a Sendmail daemon is already running. You can use either `/sbin/init.d/sendmail stop` or `killsm` to stop the running daemon.

Configuring and Reading the Sendmail Log

Sendmail logs its mail messages through the `syslogd` logging facility.

The `syslogd` configuration must write mail logging to the file `/var/adm/syslog/mail.log`. You can do this by adding the following line in `/etc/syslog.conf`:

```
mail.debug  /var/adm/syslog/mail.log
```

You can use the HP `mtail` utility to look at a specified number of the last lines of the log file:

```
mtail 15
```

By default, `mtail` displays the last 20 lines of the log file. For more information on the `mtail` utility, type `man 1M mtail` at the HP-UX prompt.

For more information on configuring `syslogd`, see the *HP-UX Internet Services Administrator's Guide* at the URL `http://www.docs.hp.com/hpux/netcom/index.html#Internet%20Services`.

Setting Log Levels

You can set the log level with the `-oL` option on the Sendmail command line or on the `OL` line in the Sendmail configuration file. At the lowest level, no logging is done. At the highest level, even the most mundane events are recorded. As a convention, log levels 11 and lower are

considered useful. Log levels above 11 are normally used only for debugging purposes. We recommend that you configure `syslogd` to log mail messages with a priority level of debug and higher. Sendmail's behavior at each log level is described in Table 2-5.

Table 2-5 Sendmail Logging Levels

Logging Level	Behavior
0	No logging.
1	Major problems only.
2	Message collections and failed deliveries.
3	Successful deliveries.
4	Messages being queued (due to a host being down, and so on).
5	Messages being added to the queue in routine circumstances.
6	Unusual but benign incidents, such as trying to process a locked queue file.
9	Log internal queue ID to external message ID mappings. This can be useful for tracing a message as it travels between several hosts.
10	The name of the mailer used, the host (if nonlocal), and the user name passed to the mailer are logged. If the log level is 10 or higher, Sendmail also reports this information in `-bv` (verify) mode.
11	For successful deliveries to IPC mailers, the `MX` (mail exchanger) host delivered to (if any) and the Internet address used for the connection are logged.
12	All incoming and outgoing SMTP commands and their arguments are logged at LOG_INFO.
13	Log bad user shells, world-writable files, and other questionable situations.

Table 2-5 Sendmail Logging Levels (Continued)

14-98	Debugging information. This information must be interpreted by your HP service representative.

Understanding syslog Entries

Sendmail logs the following:

- Failures beyond its control (`SYSERR`).
- Administrative activities (for example, rebuilding the aliases database, and killing and restarting the daemon).
- Events associated with mail transactions.

Log entries marked `SYSERR` indicate either system failures or configuration errors and may require the attention of the system administrator.

Each system log entry for a mail transaction has a queue ID associated with it. All log entries for the same input message have the same queue ID. Log level is normally set to 10 in the configuration file. At this level, the following information is logged for each delivery:

`message-id=` If a message had a `Message ID` header line when it was input to Sendmail, this is logged. Sendmail can also be configured to add a `Message ID` header line if none is present. This ID uniquely identifies a message and can be used to trace the progress of a message through mail relays.

`from=` The sender of the message and the message size are logged.

`to=` The recipient of the message. One message may have multiple recipients. Sendmail logs a separate entry for each separate delivery attempt it makes, so multiple recipients on the same host may appear on the same line, but multiple recipients on different hosts will appear on different lines. The delivery status of the message (whether message succeeded, failed, or was queued), the mailer, and the host used are logged.

Other details logged in the syslog file are time delay in delivering the message (delay=), type of mailer used (mailer=), priority of the message, relay machine, and the status of the message. Queued messages and SYSERRs are also logged.

Storing Off Old Sendmail Log Files

At typical logging levels, every piece of mail passing through Sendmail adds two or three lines to the mail log. A script to manage the growth of the mail log could be run nightly, at midnight, with an entry in root's crontab file. Following is an example of a crontab entry for a script called newsyslog:

```
0 0 * * * /var/adm/syslog/newsyslog
```

The following example shows what the script /var/adm/syslog/newsyslog might contain. The script assumes that syslog is configured to direct mail logging to /var/adm/syslog/mail.log.

```
#!/usr/bin/sh
#
#  NEWSYSLOG: Save only the last week's Sendmail logging.
cd /var/adm/syslog
#
mv mail.log.6  mail.log.7
mv mail.log.5  mail.log.6
mv mail.log.4  mail.log.5
mv mail.log.3  mail.log.4
mv mail.log.2  mail.log.3
mv mail.log.1  mail.log.2
cp mail.log    mail.log.1
kill -1 `cat /var/run/syslog.pid`
```

Printing and Reading the Mail Queue

You can print the current contents of the mail queue with the following command:

```
mailq
```

The output looks similar to this example:

```
    /var/spool/mqueue (3 requests)

----Q-ID-----   --Size---   -----Q-Time----    ----Sender/Recipient-----
```

```
h3TA9Bb29701    86          Wed Feb 9 07:08    janet
                                               ess@vetmed.umd.edu
                                               ebs@surv.ob.com

h3TAATe29713    1482        Tue Feb 15 7:05    carole
                                               bja@edp.cloq.potlatch.com
                                               vls@ee.cmu.edu

h3TABWB29731    10169       Tue Feb 15 8:10    chuck
                                               hrm@per.stmarys.com
                                               sys6!sysloc@njm
```

The first entry is a message with queue ID `h3TA9Bb29701` and a size of 86 bytes. The message arrived in the queue on Wednesday, February 9, at 7:08 a.m. The sender was `janet`. She sent a message to the recipients `ees@vetmed.umd.edu` and `ebs@surv.ob.com`. Sendmail has already attempted to route the message, but the message remains in the queue because its SMTP connection was refused. This usually means that the SMTP server is temporarily not running on the remote host, but it also occurs if the remote host never runs an SMTP server. Sendmail attempts to deliver this message the next time the mail queue is processed.

Two other messages in the queue are also routed for delivery the next time the mail queue is processed.

If `mailq` is run in verbose mode (with the `-v` option), then when it prints the queue, it will also show the priority of each queued message.

Files in the Mail Queue

The files that Sendmail creates in the mail queue all have names of the following format:

`ymdhmsrXXXXX`

where

`y` – Denotes the year

`m` – Denotes the month

`d` – Denotes the day

`h` – Denotes hour

`m` – Denotes minute,

`s` – Denotes second

`r` – Denotes a random number

`XXXXX` – Denotes a 5-digit number that is the process ID of the process creating the queue entry.

A file whose name begins with `df` is a data file. The message body, excluding the header, is kept in this file.

A file whose name begins with `qf` is a queue-control file, which contains the information necessary to process the job.

A file whose name begins with `xf` is a transcript file. This file is normally empty while a piece of mail is in the queue. If a failure occurs, a transcript of the failed mail transaction is generated in this file.

The queue-control file (type `qf`) is structured as a series of lines, each beginning with a letter that defines the content of the line. Lines in queue-control files are described in Table 2-6.

Table 2-6 Lines in Queue-Control Files

Initial Letter	Content of Line
`B`	The message body type (either `7bit` or `8bitmime`).
`C`	The controlling user for message delivery. This line always precedes a recipient line (`R`) that specifies the name of a file or program name. This line contains the user name that Sendmail must run as when it is delivering a message into a file or a program's `stdin`.
`D`	The name of the data file. There can be only one `D` line in the queue-control file.
`E`	An error address. If any such lines exist, they represent the addresses that must receive error messages.
`H`	A header definition. There can be many `H` lines in the queue-control file. Header definitions follow the header definition syntax in the configuration file.
`P`	The current message priority. This is used to order the queue. Higher numbers mean lower priorities. The priority decreases (that is, the number grows) as the message sits in the queue. The initial priority depends on the message precedence, the number of recipients, and the size of the message.

Table 2-6 Lines in Queue-Control Files (Continued)

Initial Letter	Content of Line
M	A message. This line is printed by the `mailq` command and is generally used to store status information (that is, the reason the message was queued). It can contain any text.
R	A recipient address. Normally this has already been completely aliased, but it is actually re-aliased when the queue is processed. There is one line for each recipient.
S	The sender address. There can be only one sender address line.
T	The job creation time (in seconds since January, 1970). This is used to determine when to time out the job.

The following example is a queue-control file named `qfAA00186`. The sender is `david`, and the recipient is the local user `carolyn`. The current priority of the message is 17. The job creation time, in seconds since January, 1970, is 515 961 566. The last seven lines describe the header lines that appear on the message.

```
P17
T515961566
DdfAA00186
Sdavid
Rcarolyn
Hreceived: by lab; Thu, 8 May 86 12:39:26 mdt
Hdate: Thu, 8 May 86 12:39:26 mdt
Hfrom: David <david>
Hfull-name: David
Hreturn-path: <david>
Hmessage-id: <8605081839.AA00186@lab.HP>
Happarently-to: carolyn
```

Queue Changes

The following miscellaneous enhancements have been made to the queue option:

- The queue option allows multiple `-qI`, `-qR`, or `-qS` queue run limiters.
  For example, using `Sendmail -qRfoo -qRbar` will deliver mail to recipients with `foo` or `bar` in their address.
- The map flag `-Tx` appends `x` to lookups that return temporary failure. This is similar to `-ax` flag, which appends `x` to lookups that return success.
- The `QueueSortOrder` option is case sensitive.

Changes to Sendmail Files and Databases

Sendmail files and databases are stored in the `/etc/mail` directory. Sendmail utilities access these files and databases for their operation. If you are logged in as a root user, warning messages are displayed when you run any Sendmail utility that access these files and databases. The warning messages are displayed only when the Sendmail files and databases have incorrect permission for non-root users.

This section discusses the warning messages displayed when you execute the Sendmail utilities `mailstats` and `newaliases`. This section also describes the warning messages that appear when you send mail. Finally, this section provides information on how you can resolve these warning messages.

NOTE The warning messages do not indicate any error in the syntax of the command.

The mailstats Utility

The `mailstats` utility enables you to collect the mail statistics stored in the `/etc/mail/sendmail.st` file. If you run the `mailstats` utility with root user permission, the following warning messages might appear:

```
#mailstats
warning: /etc/mail/sendmail.st has group read/write or world
read/write permission. This is unsafe
Statistics from Thu Dec 19 10:27:00 2002
```

```
M   msgsfr   bytes_from   msgsto   bytes_to   msgsrej   msgsdis   Mailer
0   0        0K           46       47K        0         0         prog
3   41       43K          56       57K        0         0         local
5   49       51K          34       34K        0         0         esmtp
=============================================================
T   90       94K          136      138K       0         0
C   90                    136                 0
```

How to Resolve the Warning Messages

To resolve these warning messages, run the following command:

```
# chmod 600 /etc/mail/sendmail.st
```

Now, if you execute the `mailstats` utility, the warning messages do not appear.

The newaliases Utility

`newaliases` rebuilds the database for the mail aliases file. If you run the `newaliases` utility with root user permission, the following warning messages might appear:

```
# newaliases
warning: /etc/mail/aliases has world read or write
permission. This is unsafe.
warning: /etc/mail/aliases.db has world read or write
permission. This is unsafe.
/etc/mail/aliases: 7 aliases, longest 9 bytes, 88 bytes total
```

How to Resolve the Warning Messages

To resolve the warning messages, run the following command:

```
# chmod 640 /etc/mail/aliases /etc/mail/aliases.db
```

Now, if you execute the `newaliases` utility, the warning messages do not appear.

How to Resolve Warning Messages When You Send Mail

Warning messages may appear when you send mail as a root user. Following is an example statement:

```
#echo "Subject: Testing" | /usr/sbin/sendmail root
warning: /etc/mail/aliases has world read or write
permission. This is unsafe.
warning: /etc/mail/aliases.db has world read or write
permission. This is unsafe.
warning: /etc/mail/sendmail.st has group read/write or world
read/write permission. This is unsafe
```

Warning messages appear only for the files that have incorrect permission. To resolve the warning messages, run the appropriate commands as described in the sections "The mailstats Utility" on page 104 and "The newaliases Utility" on page 105.

Impact on Non-Root Users

With the change in permission, non-root users cannot access the files and databases associated with Sendmail, and a `Permission denied` message appears when you run any utility that access the Sendmail files and databases.

The following messages appear when you run the `praliases` and `mailstats` utilities:

```
$ praliases
praliases: /etc/mail/aliases: open: Permission denied
$ mailstats
mailstats: /etc/mail/sendmail.st: Permission denied
```

3 Sendmail 8.13.3

This chapter discusses the new features in Sendmail 8.13.3, which is the latest Web upgrade of Sendmail available on the HP-UX 11i v1 and HP-UX 11i v2 operating systems.

This chapter discusses the following topics:

NOTE All occurrences to Sendmail in this chapter refer to Sendmail 8.13.3 unless specified explicitly.

Overview

Sendmail 8.13.3 is the latest version of Web upgrade available on the HP-UX 11i v1 and HP-UX 11i v2 operating systems at `http://www.software.hp.com`.

The main difference between Sendmail 8.11.1 and Sendmail 8.13.3 is that Sendmail 8.13.3 can act as a Mail Submission Program (MSP) using a different configuration file compared to the one used by the Sendmail daemon MTA. The `/etc/mail/submit.cf` file is the default Sendmail MSP configuration file.

When Sendmail starts up in the daemon mode, it listens both on the normal port 25 for incoming SMTP connections and on port 587 for the local submission of mail. The latter role is that of an MSA (documented in RFC 2476) and requires that Mail User Agents (MUAs) be explicitly coded to use port 587 for local submission of mail directly to the Sendmail daemon.

NOTE The role of the Sendmail daemon as an MSA (introduced on 8.11.1) is a different concept from that of the role of Sendmail as an MSP.

When Sendmail 8.13.3 is executed independently or invoked from a MUA to process locally submitted mail, Sendmail takes on the role of an MSP. MSP accepts and processes the submitted mail messages as a non-root user and queues them separately. After processing, MSP delivers the submitted mail messages to the Sendmail MTA daemon using the SMTP protocol through the port 25. The default `/etc/mail/submit.cf` file assumes that the MTA sendmail daemon is running on the local host.

When the Sendmail MTA is started, by default an additional Sendmail MSP queue-processing daemon is also started. The MSP queue daemon does not listen on any socket. The only purpose of the MSP queue daemon is to periodically scan the MSP mail queues for any mail messages accepted by MSP which have not yet been forwarded to the Sendmail MTA daemon.

New Features in Sendmail 8.13.3

This chapter discusses the following new features in Sendmail 8.13.3:

The following sections discuss the new features in detail.

LDAP Enhancements to Support Recursion and LDAP URL Support

Sendmail 8.13.3 supports LDAP recursion based on the `TYPE`s given to attribute specifications in an LDAP map definition. This allows LDAP queries to return a new query, a DN, or an LDAP URL which will in turn be queried.

LDAP recursion allows you to add `TYPE`s to the search attributes on an LDAP map specification. The syntax for LDAP recursion is as follows:

```
-v ATTRIBUTE[:TYPE[:OBJECTCLASS[|OBJECTCLASS|...]]]
```

Following are the various `TYPE`s available:

`NORMAL` This attribute type specifies the attribute to add to the results string. This is the default `TYPE` value.

DN	Any matches for this attribute are expected to have a value of a fully qualified distinguished name. Sendmail looks up that DN and applies the attributes requested to the returned DN record.
FILTER	Any matches for this attribute are expected to have a value of an LDAP search filter. Sendmail performs a lookup with the same parameters as the original search but replaces the search filter with the one specified here.
URL	Any matches for this attribute are expected to have a value of an LDAP URL. Sendmail performs a lookup of that URL and uses the results from the attributes named in that URL. Note however that the search is done using the current LDAP connection, regardless of what is specified as the scheme, LDAP host, and LDAP port in the LDAP URL.

Any untyped attributes are considered NORMAL attributes.

The optional OBJECTCLASS (| separated) list contains the objectClass values for which that attribute applies. If the list is given, the attribute named will only be used if the LDAP record being returned is a member of that object class. If these new value attribute TYPEs are used in an AliasFile option setting, it will need to be double quoted to prevent Sendmail from misparsing the colons.

LDAP recursion attributes which do not ultimately point to an LDAP record are not considered as an error.

Following is an example of LDAP recursion that uses all the four new TYPEs:

```
O LDAPDefaultSpec=-h ldap.example.com -b dc=example,dc=com

Kexample ldap
 -z,
 -k(&(objectClass=sendmailMTAAliasObject)(sendmailMTAKey=%0))
 -v sendmailMTAAliasValue,mail:NORMAL:inetOrgPerson,
    uniqueMember:DN:groupOfUniqueNames,
    sendmailMTAAliasSearch:FILTER:sendmailMTAAliasObject,
     sendmailMTAAliasURL:URL:sendmailMTAAliasObject
```

This definition specifies that:

- Any value in a `sendmailMTAAliasValue` attribute is added to the result string regardless of the object class.
- The mail attribute is added to the result string if the LDAP record is a member of the `inetOrgPerson` object class.
- The `uniqueMember` attribute is a recursive attribute, used only in `groupOfUniqueNames` records, and must contain an LDAP DN pointing to another LDAP record. The intention here is to return the mail attribute from those DNs.
- The `sendmailMTAAliasSearch` attribute and `sendmailMTAAliasURL` are used only if referenced in a `sendmailMTAAliasObject`. They are both recursive, the first for a new LDAP search string and the latter for an LDAP URL.

Support for the FallBackSmartHost Option

When Sendmail prepares to connect to a remote host for transfer of mail, it first performs a series of checks to identify the remote host. Sendmail looks up the MX records and calls the `res_search()` BIND library routine to find all MX records for the host. If Sendmail does not find the MX records, it tries to deliver the message to a single original host, which is a central mail hub to which mail can be forwarded. If this fails, Sendmail attempts to deliver to the host listed with the `FallbackMXHost` option.

Following is the format of the `FallbackMXHost` option:

```
FallbackMXhost=fallbackhost
```

The `FallbackMXhost` option works only if Sendmail can look up the host name of the recipient. If Sendmail does not find the host name, the `FallbackMXhost` is not useful. In such situations, Sendmail uses the `FallBackSmartHost` option.

The `FallBackSmartHost` option specifies the name of a mail exchange (MX record) that Sendmail must use as a last resort when MX records are not available to identify the remote host. This option is given an artificially low priority so that Sendmail tries to connect to it only if all other connection attempts for the remote host have failed.

Following is the format for the `FallBackSmartHost` option:

```
FallBackSmartHost=hostname
```

where, hostname specifies the canonical name to which the host will fallback.

Mail message forwarded to that host name fails if hostname is an empty string or is the name of a nonexistent host. You can also use macros to represent the hostname. Sendmail expands these macros before connecting to the remote host. If the hostname that you specify for the FallBackSmartHost option exists in the $=w class, Sendmail silently ignores the hostname.

The FallBackSmartHost option is also useful for unreliable FallbackMXhost servers. When the FallbackMXhost server goes down, Sendmail uses the FallBackSmartHost option and thus the flow of mail messages does not stop.

You must be careful while using the FallBackSmartHost option because if you specify this option from the command line, Sendmail can relinquish its special privileges.

Socket Maps

Sendmail 8.13.3 contains a new socket map to query maps through TCP/IP sockets.

The socket map uses a simple request or reply protocol over TCP or UNIX® domain sockets to query an external server, which can be a third party or a self-coded program. Neither the requests nor replies end with a carriage return (CR) or line feed (LF). Both the requests and replies are text based and encoded as net strings. A string "hello there" is represented as follows:

```
11:hello there
```

The request consists of the database map name and the lookup key separated by a space character, specified as follows:

```
<mapname> ‘’<key>
```

The server responds with the following status indicator and the result (if any):

```
<status> ‘’ <result>
```

The status indicator is one of the following upper case words:

OK	Specifies that the key is found and the result contains the looked-up value.

NOTFOUND	Specifies that the key is not found and the result is empty.
TEMP	Specifies that a temporary failure has occurred.
TIMEOUT	Specifies that a timeout has occurred on the server side.
PERM	Specifies that a permanent failure has occurred.

In case of an error, that is, when the status is TEMP, TIMEOUT, or PERM, the result field contains an explanatory error message.

Following are examples of the error messages in the result field:

- For a successful lookup:

  ```
  31:OK resolved.address@example.com
  ```

- When the key is not found:

  ```
  8:NOTFOUND
  ```

- When a failure occurs:

  ```
  55:TEMP this text explains that we had a temporary
  failure
  ```

The socket map uses the following syntax to specify the remote endpoint:

```
Xname {, field=value }*
```

where, name is the name of the filter and field=name pairs define attributes to the filter.

Following are the different field types:

Socket	Specifies the socket specification.
Flags	Specifies special flags for a filter.
Timeouts	Specifies timeouts for a filter.

Sendmail checks only the first character of the field name for the field type. The field name is case sensitive.

Following are different forms of socket specifications:

```
S=inet:port@host
```

```
S=inet6:port@host
```

```
S=local:path
```

The first two forms describe an IPv4 or IPv6 socket listening on a certain port at a given host or IP address. The last form describes a named socket on the file system at the given path.

Following is an example of a socket map that specifies a remote endpoint:

```
KmySocketMap socket inet:12345@127.0.0.1
```

If multiple socket maps define the same remote endpoint, they share a single connection to this endpoint.

DNS Maps

The `dns` map is an internal database map available to perform DNS lookups. You can use the following `K` configuration command to declare the `dns` map:

`K`*dnslookup* `dns -R`*lookup-type*

where *dnslookup* specifies the name of the map using DNS.

The `dns`-type database map is primarily used for `dnsbl` and `endnsbl` features.

You must always include the `-R` switch, which specifies the DNS resource record type to lookup, in the `dns` map declaration.

Sendmail 8.13.3 supports the following types of resource records: `A`, `AAAA`, `AFSDB`, `CNAME`, `MX`, `NS`, `PTR`, `SRV`, and `TXT`. A map lookup returns only one record. For certain types of records, such as MX records, the return value can be a random element of the list due to randomizing in the DNS resolver.

Table 3-1 describes the different `-R` values in the `dns` database map.

Table 3-1 Supported DNS Queries

-R Value	Description
`A`	Returns IPv4 address records for the host (RFC 1035)
`AAAA`	Returns IPv6 address records for the host (RFC 1886)
`AFSDB`	Returns an AFS server resource record (RFC 1183)
`CNAME`	Returns the canonical name for the host (RFC 1035)
`MX`	Returns the best MX record for the host (RFC 1035)

Table 3-1 Supported DNS Queries (Continued)

-R Value	Description
NS	Returns a name server record (RFC 1035)
PTR	Returns the host name that corresponds to an IP record (RFC 1035)
SRV	Returns the port to use for a service (RFC 2782)
TXT	Returns general (human-readable) information (RFC 1035)

To make the dns database-map more useful, you can also use the switches described in Table 3-2.

Table 3-2 The dns Database-Map Type K Command Switches

Switch	Description
-A	Appends values for duplicate keys.
-a	Appends tag on successful match.
-d	Denotes the res_search()_res.retry interval.
-f	Informs Sendmail not to fold keys to lowercase.
-m	Suppresses replacement on match.
-N	Appends a null byte to all keys.
-O	Specifies Sendmail not to add a null byte.
-o	Specifies an optional database map.
-q	Informs Sendmail not to strip quotes from the key.
-R	Specifies the record type to look up.
-r	Denotes the rs_search()_res.retries limit.
-T	Denotes the suffix to append on temporary failure.
-t	Informs Sendmail to ignore temporary errors.

Support for Deliver By SMTP Extension (RFC 2852)

The Delivery By SMTP extension is a mechanism by which an SMTP client can request a server to deliver the message within a prescribed period of time, while transmitting a message to an SMTP server. A client that makes such a request also specifies message handling which must occur if the message cannot be delivered within the specified time period. The options can be either to return the message as an undeliverable message with no further processing or to issue a delayed delivery status notification (DSN).

Following is the declaration for the Delivery By SMTP extension in the Sendmail 8.13.3 configuration file:

```
#O DeliverByMin=0
```

A value of 0 (zero) indicates that the `DeliverByMin` option is disabled. Do not consider this extension as a vehicle for requesting “priority” processing. A receiving SMTP server can assign processing priority to a message transmitted with a Delivery By request. A Delivery By request serves to express the urgency of a message and to provide an additional degree of determinancy in its processing. The message can be withdrawn if it is not delivered within the specified period of time.

A typical usage of this mechanism is to prevent delivery of a message beyond some future time of significance to the sender or recipient but not known by the MTAs handling the message.

Another common usage arises when a sender wishes to be alerted to delivery delays. In this case, the sender can mark a message such that if it is not delivered, for example within 30 minutes, a "delayed" DSN is generated but delivery attempts are nonetheless continued. In this case, senders are allowed to express a preference for when they would like to learn of delivery problems.

Anti-Spamming Features

In addition to the anti-spamming features provided by Sendmail 8.11.1, Sendmail 8.13.3 provides the following anti-spamming features:

- Message quarantining
- Support for mailer filter (MILTER) APIs for advanced and effective mail filtering
- Enhanced DNS Black Hole List (EDNSBL) option

The following sections discuss the anti-spamming features in detail.

Message Quarantining

Starting with Sendmail 8.13.3, you can quarantine mail messages, which are otherwise known as envelopes. Queue files or envelopes are stored but not considered for delivery or display unless the "quarantine" state of the envelope is undone, or delivery or display of the quarantined items is requested.

Quarantined messages are tagged using the name `hf` for the queue file instead of the name `qf` for the queue file, and by adding the quarantine reason to the queue file.

When you run the following command, the quarantine reason is displayed in a new line prefixed with `QUARANTINE`:

```
mailq -qQ
```

where, the `-qQ` option specifies the quarantined queue items.

Quarantined messages are not run on normal queue displays. They run unless specifically requested with the `-qQ` option.

You can run and display restricted mail queues based on the quarantined reason using the `-qQtext` option if the quarantine reason contains the given text. Similarly, the `-q!Qtext` runs or displays quarantined items which do not have the given text in the quarantine reason.

You can use the `-qQ flag` option to request the delivery or display of quarantined items. Additionally, you can quarantine or unquarantine messages already in the queue using the new `-Q` flag to Sendmail. For example, the following command quarantines the normal queue items matching the criteria specified by the `-q[!][I|R|S|G][matchstring]` option using the reason given in the `-Q` flag:

```
sendmail -Qreason -q[!][I|R|S|G][matchstring]
```

Similarly, you can use the following command to change the quarantine reason for the quarantined items matching the criteria specified by the `-q[!][I|R|S|Q][matchstring]` option using the reason given on the `-Q` flag:

```
sendmail -qQ -Q[reason] -q[!][I|R|S|Q|G][matchstring]
```

If you do not specify a reason, unquarantine the matching items and make them normal queue items. The `-qQ` flag informs Sendmail to operate on quarantined items instead of normal items.

A new error code for the `$#error $@ quarantine $: reason`, can be used to quarantine message in `check_* (except check_compat)` and header check rulesets. The `$:` of the mailer triplet will be used for the quarantine reason.

Support for Mail Filter (MILTER) APIs

Beginning with Sendmail 8.13.3, you can use the Mail Filter (Milter) APIs to filter all inbound messages through an external filter program. Milter is designed to allow third-party programs to access mail messages as they are being processed in order to filter meta information and content. Milter is declared in the configuration file as:

```
Xname {, field=value}*
```

where `name` is the name of the filter (used internally only) and the `field=value` pairs define attributes of the filter.

For more information on Milter, refer to the *Sendmail 8.13.3 Programer's Guide* at `http://www.docs.hp.com/en/netcom.html#Internet%20Services`.

Enhanced DNS Black Hole List Option

The enhanced DNS Black Hole List (EDNSBL) option is an enhanced version of the `dnsbl` feature.

The `dnsbl` feature rejects mail from hosts in a DNS-based rejection list. The `dnsbl` feature is used to enable the blocking of email from open relay sites, dialup sites, or known spamming sites. This feature is included in the `sendmail.cf` configuration file as:

```
# map for DNS based blacklist lookups
Kdnsbl dns -R A -T<TMP>
```

The enhanced `dnsbl` feature is a superset of the `dnsbl` feature. This feature is represented in the sendmail.cf file as follows:

```
# map for enhanced DNS based blacklist lookups
Kenhdnsbl dns -R A -a. -T<TMP> -r5
```

You must use the `/usr/newconfig/etc/mail/cf/cf/gen_cf` script to include the `enhdnsbl` feature in the `sendmail.cf` file. You must choose the "5: Enhanced DNSBL" sub-menu option in the "3: Anti-Spamming Options" main menu option, and regenerate the `sendmail.cf` file.

You can use the `dns`-type database map for the `dnsbl` and `enhdnsbl` features.

The enhancement consists of additional arguments, that is, one or more literal addresses you expect returned when an address must be rejected.

Compared to the `dnsbl` option, you can specify additional arguments (upto 5) to specify the return values from lookups. Sendmail ignores temporary lookup failures in the absence of a third argument, which must be either `t` or a full error message. By default, any successful lookup generates an error. Otherwise, the result of the lookup is compared with the supplied arguments, and only if a match occurs an error is generated.

Queuing

Starting with Sendmail 8.13.3, you can define queues according to selected criteria and process each group with custom settings. The rule sets then select the queue group to which the message of a recipient must belong.

You can use the `-q` command-line option to specify which queue to display. This is an option available in earlier versions of Sendmail. Sendmail 8.13.3 has a few additional queue-related options, such as processing only the quarantined items.

The Default Queue Group

Sendmail 8.13.3 offers a method to define multiple queue directories and a method to group them by function or specialty. For compatibility with older versions of Sendmail, there is a special queue group called `mqueue`. This is the default queue group. It takes on all the properties of every `-q` command, and every queue option.

When you declare additional queue groups, they take all their properties from the default group, unless you override a particular property with a specific equate. Table 3-3 describes the equates and the command-line arguments or options they override.

Table 3-3 Q Configuration Command Equates

Equate	Overrides Command-Line Switch/Option	Description
`Flags=(F=)`	`-qf`	Specifies fork queue runs.

Table 3-3 Q Configuration Command Equates (Continued)

Equate	Overrides Command-Line Switch/Option	Description
Interval=(I=)	-qInterval	Specifies interval between queue runs.
Jobs=(J=)	MaxQueueRunSize	Specifies the maximum number of envelopes per queue run.
Nice=(N=)	NiceQueueRun	Specifies how to renice(3) the queue run.
Path=(P=)	QueueDirectory	Specifies the queue directory or directories.
recipients=(r=)	MaxRecipientsPerMessage	Specifies the maximum recipients per envelope.
Runners=(R=)	MaxRunnersPerQueue	Specifies the maximum queue processors per queue group.

The Q Configuration Command

You can define queue groups using the Q configuration command, which specifies the name of the queue group and a sequence of equates. Following is the syntax for the Q command:

Q*groupname*, *equates*

You must not insert a space between Q and the *groupname*. You can optionally specify the equates, but if they are present they must follow the name of the queue group and they must be separated with a comma or whitespace, or both.

The equates are formed by selecting one of the keywords shown in the first column in Table 3-3, and by following the keyword with an equal sign and the value you wish to assign to that key letter. Sendmail reads only the first letter. Therefore, you can use the shorthand shown in parenthesis in Table 3-3. The first letter is case sensitive, that is, R and r are different.

For example, the following commands declare a queue directory (the `Patch=` and `P=`), and a queue processing interval of 10 minutes (the `Interval=` and `I=`):

```
Qslowmail, Path=/disk1/mail/slowqueues, Interval=10m
Qslowmail, P=/disk1/mail/slowqueues, I=10m
```

Using queuegroups Through the access Database

You must use the `gen_cf` main menu option to utilize the queuegroup feature to easily select queue groups based on recipient addresses or recipient domains.

After enabling the `queuegroup` feature, the next step is to add lines such as the following to the source file for your access database:

```
QGRP:slow-poke.com                slowgroup
QGRP:root@notify.com              fastgroup
QGRP:your.domain                     localgroup
```

Queue Group Limitations

You can define the default group (`mqueue`) using options and the command line. If a `Q` configuration command is missing a given equate, that queue group inherits the property defined by the default queue group. However, following are the default queue group properties, which do not have equivalent equates and all queue groups inherit these properties:

- `DeliveryMode` *option*
- `FastSplit` *option*
- `MaxQueueChildren` *option*
- `MinQueueAge` *option*
- `-qI`, `-qR`, and `-qS` command-line switches
- `QueueFactor`, `QueueLA`, `RefuseLA` and `RecipientFactor` options
- `QueueFileMode` *option*
- `Timeout, queuereturn` and `Timeout.queuewarn` options

You cannot override these properties with a queue-group equate.

Performance Features

Sendmail 8.13.3 contains the following performance enhancement features:

The following sections discuss the Sendmail 8.13.3 performance features in detail.

The FastSplit Option

You can use the `FastSplit` option to suppress MX lookups before splitting an envelope and to limit the number of envelopes that can be delivered on the initial attempt. The `FastSplit` option syntax is as follows:

```
-OFastSplit=num
```

where, `num` is of type numeric.

If `num` is a negative nonnumeric value, or zero, Sendmail enforces initial sorting based on MX records.

If `num` is set to a value greater than zero, the initial MX lookups on addresses are suppressed when they are sorted which may result in faster envelope splitting. If the mail is submitted directly from the command line, then the value also limits the number of processes to deliver the envelopes.

When Sendmail expands an alias, as when using aliases to send to a mailing list, Sendmail sorts the list of new recipients by host. Normally, the list of hosts is then sorted by MX records rather than host name. After sorting, the new MX-sorted list is split by Sendmail into multiple envelopes.

Envelope splitting creates multiple envelopes when there is originally only one. Each new envelope contains fewer envelope recipients. Normally, all these envelopes are delivered in parallel for delivery efficiency.

SMTP Pipelining

This feature is an extension to the SMTP service whereby a server can indicate the extent of its ability to accept multiple commands in a single TCP send operation. Using a single TCP send operation for multiple commands improves SMTP performance. SMTP pipelining is an implementation of RFC 1854 (*SMTP Service Extension for Command Pipelining*).

Connection Caching

When processing the queue, Sendmail tries to keep the last few open connections open to avoid startup and shutdown costs. This only applies to IPC and LPC connections.

When trying to open a connection, the cache is first searched. If an open connection is found, it is probed to see if it is still active by sending a `RSET` command. If this fails, it is not considered as an error; instead, the connection is closed and reopened.

The following parameters control the connection cache:

- The `ConnectionCacheSize(k)` option defines the number of simultaneous open connections that are permitted. If it is set to 0 (zero), connections will be closed as quickly as possible. This value limits the amount of system resources that Sendmail will use during queue runs. The default value is one. You must set this value appropriately for your system size. Do not set `ConnectionCacheSize` to a value greater than 4.
- The `ConnectionCacheTimeout(K)` option specifies the maximum time that any cached connection will be permitted to remain idle. When the idle time exceeds this value, the connection is closed. This number must be small (less than 10 minutes) to prevent you from grabbing too many resources from other hosts. The default `ConnectionCacheTimeout` value is 5 minutes.

Sendmail 8.13.3 Security

By default, Sendmail is a set-user-ID program. You can set it to a set-group-ID program by creating a new user smmsp and by using the `submit.cf` configuration file. If sendmail is called for initial delivery, you must use the submit.cf file with a fallback of `sendmail.cf` as configuration file.

A Mail Submission Program (MSP) is another instance of Sendmail that is used for initial mail submission. MSP uses the `/etc/mail/submit.cf` file as the configuration file. Sendmail 8.13.3 acts as an MSA or MTA depending on the operational mode.

The default configuration starting with Sendmail 8.13.3 uses one sendmail binary which acts differently based on the operation mode and supplied options.

For security reasons, Sendmail must be a set-group-ID program to allow for queuing mail in a group-writable directory. When Sendmail runs as a set-group-ID program, the default group is smmsp and the group ID is 25.

The sendmail.cf configuration file is required for Sendmail to run as a server and submit.cf configuration file is required to run Sendmail as a mail submission program.

You must use the following permissions for the Sendmail configuration and default queue files:

- `-r-xr-sr-x root smmsp ... /PATH/TO/sendmail`

  This denotes that the owner of sendmail is `root`, the group is `smmsp`, and the binary is set-group-ID.

- `drwxrwx--- smmsp smmsp ... /var/spool/clientmqueue`

  This denotes that the client mail queue is owned by `smmsp` with group `smmsp` and is group writable. The client mail queue directory must be writable by `smmsp`. In the `submit.cf` file, you must set the `UseMSP` option and you must set the `QueueFileMode` option to 0660.

- `drwx------ root wheel ... /var/spool/mqueue`
- `-r--r--r-- root wheel ... /etc/mail/sendmail.cf`
- `-r--r--r-- root wheel ... /etc/mail/submit.cf`

This section discusses the following topics:

- "Support for Secured Mail Transaction using STARTTLS" on page 126
- "Cyrus SASL v2 Support" on page 127
- "The submit.cf File" on page 128

Support for Secured Mail Transaction using STARTTLS

STARTTLS is the SMTP command to "Start Transport Layer Security"; or in other words to turn on Secure Socket Layer (SSL). Transport Layer Security (TLS) provides authentication (identification), privacy, confidentiality, and integrity for securing a mail transaction. TLS uses different STARTTLS algorithms for encryption, signing, and message authentication.

The STARTTLS configuration uses the following variables:

`UseTLS` — Enables the TLS handshake in the SMTP transaction. You can set this variable to either `True` or `False`. Following is the option in the `sendmail.cf` file:

```
# O UseTLS=False
```

`CERT_DIR` — Specifies the directory for storing Sendmail certificates. Following is the option in the `sendmail.cf` file:

```
# CA directory
O CACertPath=/etc/mail/certs/
```

`CACERT_PATH` — Specifies the path that stores the certificates of all the Certificate Authorities known to the Sendmail server.

`CACERT` — Specifies the file containing the certificate of the Certificate Authority that issued the certificate of the Sendmail server.

`SERVER_CERT` and `CLIENT_CERT` — Refers to the server and client certificate. These variables indicate that the certificate of the server is used when acting as a server and when acting as a client. Following is the option in the `sendmail.cf` file:

```
# Server Cert
OServerCertFile=/etc/mail/certs/oldcert.pem
# Client Cert
OClientCertFile=/etc/mail/certs/oldcert.pem
```

`SERVER_KEY` and `CLIENT_KEY` — Specifies the private keys that correspond to the certificates of the Sendmail server. Following is the option in the `sendmail.cf` file:

```
# Server private key
O ServerKeyFile=/etc/mail/certs/oldreq.pem
# Client private key
O ClientKeyFile=/etc/mail/certs/oldreq.pem
```

You can use the `/usr/newconfig/etc/mail/cf/cf/gen_cf` script to generate the `sendmail.cf` configuration file that supports the STARTTLS feature. The generated configuration file contains all the STARTTLS options discussed previously. But, these options contain default values and are commented by default. The `gen_cf` script gives you the option to change the default values. If you change the default values for a particular option, the option is enabled or uncommented in the generated `sendmail.cf` configuration file.

To use Sendmail with STARTTLS, you must install the OpenSSL software on your system from `http://www.software.hp.com`.

Cyrus SASL v2 Support

The Simple Authentication and Security Layer (SASL), is a generic mechanism for protocols to accomplish authentication. Because protocols (such as SMTP or IMAP) use SASL, it is a natural place for code sharing between applications. Some notable applications that use SASL include Sendmail and Cyrus `imapd` (versions 1.6.0 and higher).

Applications use the SASL library to inform it how to accomplish the SASL protocol exchange, and what the results are.

SASL is only a framework and specific SASL mechanisms govern the exact protocol exchange. If there are `n` protocols and `m` different ways of authenticating, SASL attempts to make it so only `n` plus `m` different specifications need be written instead of `n` times `m` different specifications. With the Cyrus SASL library, the mechanisms need only be written once, and they work with all servers that use it.

How SASL Works How SASL works is governed by what mechanism the client and server choose to use and the exact implementation of that mechanism. This section describes the way these mechanisms act in the Cyrus SASL implementation.

The PLAIN Mechanism and sasl_checkpass() Call The PLAIN mechanism is not a secure method of authentication by itself. It is intended for connections that are being encrypted by another level. For example, the IMAP command "STARTTLS" creates an encrypted connection over which PLAIN can be used. The PLAIN mechanism works by transmitting a `user ID`, an authentication `ID`, and a password to the server, and the server then determines whether that is an allowable triple.

The principal concern is how the authentication and password are verified. The Cyrus SASL library is flexible in this regard.

A standard Cyrus SASL configuration file looks like:

```
srvtab: /var/app/srvtab
pwcheck_method: kerberos_v4
```

Application Configuration Applications can redefine how the SASL library looks for configuration information.

For instance, Cyrus `imapd` reads its SASL options from its own configuration file, `/etc/imapd.conf`, by prepending all SASL options with `sasl_`: The SASL option `pwcheck_method` is set by changing `sasl_pwcheck_option` in the `/etc/imapd.conf` file.

Configuring Cyrus SASL v2 in Sendmail To configure Cyrus SASL v2 in Sendmail, you must change the default values for the following options in the Sendmail configuration file:

```
C{TrustAuthMech}GSSAPI DIGEST-MD5 CRAM-MD5 ANONYMOUS PLAIN

# list of authentication mechanisms
O AuthMechanisms=EXTERNAL GSSAPI KERBEROS_V4 DIGEST-MD5
CRAM-MD5 ANONYMOUS PLAIN

# Authentication realm
#O AuthRealm

# default authentication information for outgoing connections
O DefaultAuthInfo=/etc/mail/default-auth-info
```

The submit.cf File

The `submit.cf` file is the client configuration file for Sendmail. The `/usr/newconfig/etc/mail/cf/cf/submit.cf.gen` file is the default Sendmail configuration file. You can also use the `/usr/newconfig/etc/mail/cf/cf/gen_cf` script to regenerate the

`submit.cf.gen` file in the `/usr/newconfig/etc/mail/cf/cf/` directory. You must copy the `/usr/newconfig/etc/mail/cf/cf/submit.cf.gen` file to the `/etc/mail` directory as `submit.cf`.

New Menu Options in the gen_cf Script

Sendmail 8.13.3 contains the following new menu options in the `/usr/newconfig/etc/mail/cf/cf/gen_cf` script:

- Create User and Queue for MSP
- Correct permissions for the sendmail files
- Verify permissions for the sendmail files
- Enhanced DNSBL
- Milter: Modify (Add/Remove/List) Filters
- Queue Groups

These new menu options are discussed in the subsections under the section "New Features in Sendmail 8.13.3" on page 110 and "The /usr/newconfig/etc/mail/cf/cf/gen_cf Script" on page 129.

The /usr/newconfig/etc/mail/cf/cf/gen_cf Script

Compared to the `/usr/newconfig/etc/mail/cf/cf/gen_cf` script menu options in Sendmail 8.11.1, the main menu options in the `/usr/newconfig/etc/mail/cf/cf/gen_cf` script in Sendmail 8.13.3 are rearranged as follows:

1. General Features
2. Relay Options
3. Anti-Spamming Options
4. Security Options
5. Generate sendmail.cf
6. Generate submit.cf
7. Verify permissions for the sendmail files
8. Correct permissions for the sendmail files
9. Create User and Queue for MSP
10. Help

You can select the relevant option to display the submenu options. The following discusses the main menu options in detail:

- The “General Features” main menu option contains the following submenu options:

  1. Delay checks
  2. No default MSA
  3. LDAP Routing
  4. Mailertable
  5. Genericstable
  6. Domaintable
  7. Virtusertable
  8. Send only
  9. Receive only
  10. Queue Groups
  11. Accept unresolvable domains
  12. Accept unqualified senders

  You can select the relevant submenu option to set the appropriate options in the `/usr/newconfig/etc/mail/cf/cf/sendmail.cf.gen` file.

- The “Relay Options” main menu option contains the following submenu options:

  1. Relay ON
  2. Relay OFF [Default Sendmail.cf]
  3. Relay entire domain
  4. Relay based on MX
  5. Relay hosts only
  6. Relay local from
  7. Loose relay check
  8. Promiscuous relay
  9. Relay mail from

You can select the relevant submenu option to set the appropriate relay options in the `/usr/newconfig/etc/mail/cf/cf/sendmail.cf.gen` file.

- The “Anti Spamming Options” main menu option contains the following submenu options:

  1. Access DB
  2. Blacklist Recipients
  3. RBL
  4. DNSBL
  5. Enhanced DNSBL
  6. Milter: Modify (Add/Remove/List) filters

  You can select the relevant submenu option to set the appropriate anti-spamming options in the `/usr/newconfig/etc/mail/cf/cf/sendmail.cf.gen` file.

- The “Security Options” main menu option contains the following submenu options:

  1. Smrsh
  2. STARTTLS

  You can select the relevant submenu option to set the appropriate security options in the `/usr/newconfig/etc/mail/cf/cf/sendmail.cf.gen` file.

- The “Generate sendmail.cf” menu option generates `sendmail.cf.gen` file in the `/usr/newconfig/etc/mail/cf/cf` directory. You must copy the `/usr/newconfig/etc/mail/cf/cf/sendmail.cf.gen` file as `/etc/mail/sendmail.cf` file.

- The “Generate submit.cf” menu option generates the `submit.cf.gen` file in the `/usr/newconfig/etc/mail/cf/cf` directory. You must copy the `/usr/newconfig/etc/mail/cf/cf/submit.cf.gen` file as `/etc/mail/submit.cf` file.

- The “Verify permissions for the sendmail files” menu option verifies the permission of the Sendmail files. In Sendmail 8.11.1, you could use the `gen_cf` command-line option `-v` to verify the permissions. Starting with Sendmail 8.13.3, you can also use the `gen_cf` script to verify the permissions of the Sendmail files.
- The “Correct permissions for the sendmail files” menu option corrects the permissions of the Sendmail files. Previously, you could use the `gen_cf` command-line option `-u` to correct the permissions. Starting with Sendmail 8.13.3, you can use the `gen_cf` script to verify the permissions of the Sendmail files.
- The “Create User and Queue for MSP” menu option creates a user and queue for MSP.

NOTE For more information on the `gen_cf` submenu options, you can choose the “10. Help” main menu option.