Managing sendmail Services in Oracle® Solaris 11.1

Part No: E28999–01
October 2012

121010@25097

Contents

Figures

Tables

Examples

Preface

Managing sendmail Services in Oracle Solaris 11.1 is part of a multivolume set that covers a significant part of the Oracle Solaris system administration information. This book assumes that you have already installed the Oracle Solaris operating system, and you have set up any networking software that you plan to use.

Note – This Oracle Solaris release supports systems that use the SPARC and x86 families of processor architectures. The supported systems appear in the *Oracle Solaris OS: Hardware Compatibility Lists*. This document cites any implementation differences between the platform types.

Who Should Use This Book

This book is intended for anyone responsible for administering one or more systems that run the Oracle Solaris release. To use this book, you should have one to two years of UNIX system administration experience. Attending UNIX system administration training courses might be helpful.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P–1 Typographic Conventions

Typeface	Description	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`

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

aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX system prompt and superuser prompt for shells that are included in the Oracle Solaris OS. Note that the default system prompt that is displayed in command examples varies, depending on the Oracle Solaris release.

TABLE P–2 Shell Prompts

Shell	Prompt
Bash shell, Korn shell, and Bourne shell	`$`
Bash shell, Korn shell, and Bourne shell for superuser	`#`
C shell	`machine_name%`
C shell for superuser	`machine_name#`

CHAPTER 1

Mail Services (Overview)

Setting up and maintaining an electronic mail service involves complex tasks that are critical to the daily operation of your network. As a network administrator, you might need to expand an existing mail service. Alternately, you might need to set up a mail service on a new network or a subnet. The chapters on mail services can help you plan and set up a mail service for your network. This chapter provides links to descriptions of new features in `sendmail`, as well as a list of other sources of information. The chapter also provides overviews of the software and hardware components that are required to establish a mail service.

- "What's New With Mail Services" on page 15
- "Other `sendmail` Information Sources" on page 17
- "Introduction to the Components of Mail Services" on page 17

See Chapter 2, "Mail Services (Tasks)," for procedural information about how to set up and administer mail services. For details, refer to "Task Map for Mail Services" on page 21.

See Chapter 3, "Mail Services (Reference)," for a more detailed description of the components of mail services. This chapter also describes the mail service programs and files, the mail routing process, the interactions of `sendmail` with name services, and the features in version 8.13 of `sendmail`. See "Changes in Version 8.13 of `sendmail`" on page 97.

What's New With Mail Services

This section provides information about new features in various Oracle Solaris releases.

Changes in this Release

The following changes have been made in the Oracle Solaris 11 release:

- The default version of sendmail has been updated to 8.14.5.
- The sendmail instance was split into two instances to provide better management of the traditional daemon (svc:/network/smtp:sendmail) and the client queue runner (svc:/network/smtp:sendmail-client).
- The system can be configured to automatically rebuild the sendmail.cf and the submit.mc configuration files. The required steps are documented in "How to Automatically Rebuild a Configuration File" on page 35.
- By default, the sendmail daemon runs in the new local daemon mode. The local–only mode only accepts incoming mail from the local host or loopback SMTP connections. For instance, mail from a cron job or between local users would be accepted. Outbound mail is routed as expected, only the incoming mail is changed. The -bl option is used to select the local-only mode, also known as the Become Local mode. For more information about this mode, see the sendmail(1M) man page. For instructions on how to change back to the -bd or Become Daemon mode, see "How to Use sendmail in the Open Mode" on page 36.
- The -t and -u options to the makemap command now work as expected. The delimiter declared with the -t option is used as the delimiter, even with the -u option. Previously a space would be used as a delimiter if the -u option was used, regardless of the delimiter defined by the -t option. See the makemap(1M) man page for more information about these options.

Significant Changes in Earlier Releases

- sendmail supports SMTP using Transport Layer Security (TLS). For more information, see the following:
  - "Support for Running SMTP With TLS in Version 8.13 of sendmail" on page 97
  - "How to Set SMTP to Use TLS" on page 37
- sendmail version 8.13 was added. For information about version 8.13 and other changes, see the following:
  - "Flags Used and Not Used to Compile sendmail" on page 64
  - "MILTER, Mail Filter API for sendmail" on page 65
  - "Versions of the Configuration File" on page 66
  - "Enhancement for vacation Utility" on page 77
  - "Contents of the /etc/mail/cf Directory" on page 79
  - "Changes in Version 8.13 of sendmail" on page 97
  - "Support for TCP Wrappers From Version 8.12 of sendmail" on page 106

- The mail service is managed by the Service Management Facility. Administrative actions on this service, such as enabling, disabling, or restarting, can be performed by using the `svcadm` command. The service's status can be queried by using the `svcs` command. For more information about the Service Management Facility, see the `smf(5)` man page and Chapter 2, "Managing Services (Overview)," in *Managing Services and Faults in Oracle Solaris 11.1*.

Other `sendmail` Information Sources

The following is a list of additional information sources about `sendmail`.

- Costales, Bryan. *sendmail, Third Edition*. O'Reilly & Associates, Inc., 2002.
- Home page for `sendmail` – `http://www.sendmail.org`.
- FAQ for `sendmail` – `http://www.sendmail.org/faq`.
- README for new `sendmail` configuration files – `http://www.sendmail.org/m4/readme.html`.
- A guide for issues that are related to migrating to more recent versions of `sendmail` – `http://www.sendmail.org/vendor/sun/`.

Introduction to the Components of Mail Services

Many software and hardware components are required to establish a mail service. The following sections give a quick introduction to these components. These sections also provide some of the terms that are used to describe the components.

The first section, "Overview of the Software Components" on page 17, defines the terms that are used when discussing the software parts of the mail delivery system. The next section, "Overview of the Hardware Components" on page 18, focuses on the functions of the hardware systems in a mail configuration.

Overview of the Software Components

The following table introduces some of the software components of a mail system. Refer to "Software Components" on page 67 for a complete description of all of the software components.

Component	Description
`.forward` files	Files that you can set up in a user's home directory to redirect mail or to send mail to a program automatically

Component	Description
mailbox	A file on a mail server that is the final destination for email messages
mail addresses	Address that contains the name of the recipient and the system to which a mail message is delivered
mail aliases	An alternate name that is used in a mail address
mail queue	A collection of mail messages that needs to be processed by the mail server
postmaster	A special mail alias that is used to report problems and to ask questions about the mail service
sendmail configuration file	A file that contains all the information necessary for mail routing

Overview of the Hardware Components

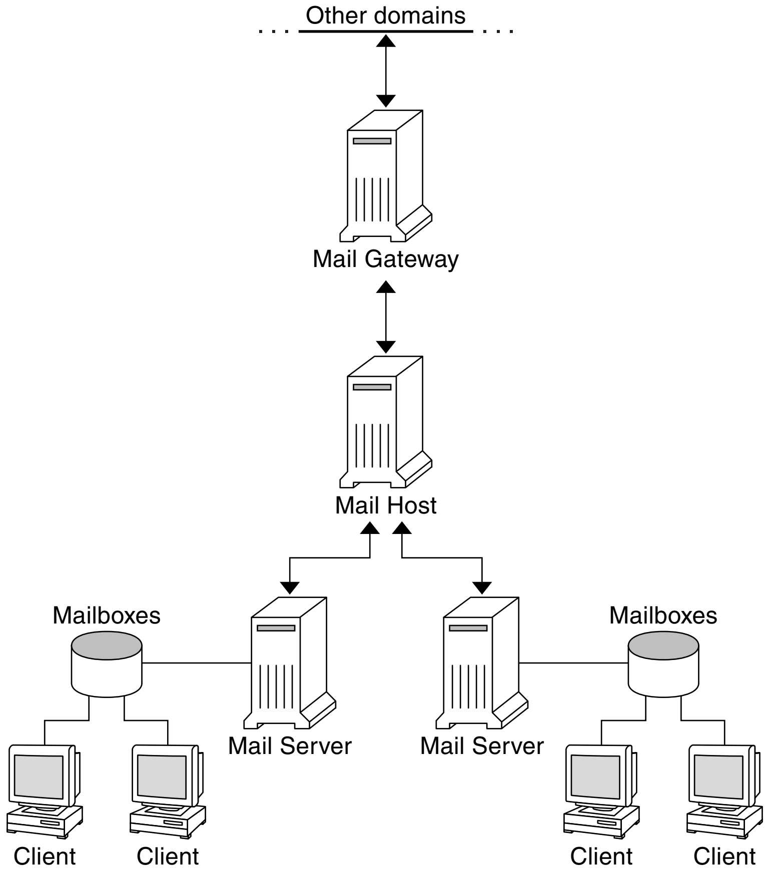
A mail configuration requires three elements, which you can combine on the same system or provide in separate systems.

- A mail host – A system that is configured to handle email addresses that are difficult to resolve
- A minimum of one mail server – A system that is configured to hold one or more mailboxes
- Mail clients – Systems that access mail from a mail server

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

Figure 1–1 shows a typical electronic mail configuration, using the three basic mail elements plus a mail gateway.

FIGURE 1–1 Typical Electronic Mail Configuration



Each element is described in detail in "Hardware Components" on page 74.

CHAPTER 2

Mail Services (Tasks)

This chapter describes how to set up and administer mail services. If you are not familiar with administering mail services, read Chapter 1, "Mail Services (Overview)," for an introduction to the components of mail services. This chapter also provides a description of a typical mail service configuration, as shown in Figure 1–1. The following list can help you find groups of related procedures that are covered in this chapter.

- "Task Map for Mail Services" on page 21
- "Setting Up Mail Services (Task Map)" on page 25
- "Changing the `sendmail` Configuration (Task Map)" on page 33
- "Administering Mail Alias Files (Task Map)" on page 42
- "Administering the Queue Directories (Task Map)" on page 48
- "Administering `.forward` Files (Task Map)" on page 52
- "Troubleshooting Procedures and Tips for Mail Services (Task Map)" on page 54

See Chapter 3, "Mail Services (Reference)," for a more detailed description of the components of mail services. This chapter also describes the mail service programs and files, the mail routing process, the interactions of `sendmail` with name services, and the features in version 8.13 of `sendmail` that are not fully described in the `sendmail`(1M) man page.

Task Map for Mail Services

The following table refers you to other task maps that focus on a specific group of procedures.

Task	Description	For Instructions
Setting up mail services	Use these procedures to set up each component of your mail service. Learn how to set up a mail server, a mail client, a mail host, and a mail gateway. Learn how to use DNS with `sendmail`.	"Setting Up Mail Services (Task Map)" on page 25

Task	Description	For Instructions
Altering the sendmail configuration	Use these procedures to modify your configuration files or service properties.	"Changing the sendmail Configuration (Task Map)" on page 33
Administering mail alias files	Use these procedures to provide aliasing on your network. Learn how to set up an NIS map, a local mail alias, a keyed map file, and a postmaster alias.	"Administering Mail Alias Files (Task Map)" on page 42
Administering the mail queue	Use these procedures to provide smooth queue processing. Learn how to display and move the mail queue, force mail queue processing, and run a subset of the mail queue. Also, learn how to run the old mail queue.	"Administering the Queue Directories (Task Map)" on page 48
Administering .forward files	Use these procedures to disable .forward files or change the search path of the .forward file. Also, learn how to permit users to use the .forward file by creating and populating /etc/shells.	"Administering .forward Files (Task Map)" on page 52
Troubleshooting procedures and tips for mail services	Use these procedures and tips to resolve problems with your mail service. Learn how to test the mail configuration, check mail aliases, test the sendmail rule sets, verify connections to other systems, and log messages. Also, learn where to look for other mail diagnostic information.	"Troubleshooting Procedures and Tips for Mail Services (Task Map)" on page 54
Resolving error messages	Use the information in this section to resolve some mail-related error messages.	"Resolving Error Messages" on page 59

Planning Your Mail System

The following list describes some concerns that should be part of your planning process.

- Determine the type of mail configuration that meets your requirements. This section describes two basic types of mail configuration and briefly lists what you need to set up each configuration. If you need to set up a new mail system or if you are expanding an existing one, you might find this section useful. "Local Mail Only" on page 23 describes the first configuration type, and "Local Mail and a Remote Connection" on page 24 describes the second type.
- As necessary, choose the systems that are to act as mail servers, mail hosts, and mail gateways.

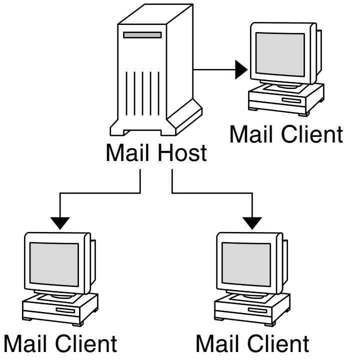
- Make a list of all the mail clients for which you are providing service and include the location of their mailboxes. This list can help you when you are ready to create mail aliases for your users.
- Decide how to update aliases and forward mail messages. You might set up an `aliases` mailbox as a place for users to send requests for mail forwarding. Users could also use this mailbox to send requests for changes to their default mail alias. If your system uses NIS, you can administer mail forwarding, rather than requiring users to manage mail forwarding. "Administering Mail Alias Files (Task Map)" on page 42 provides a list of tasks that are related to aliasing. "Administering `.forward` Files (Task Map)" on page 52 provides a list of tasks that are related to managing `.forward` files.

After you have completed the planning process, set up the systems on your site to perform the functions that are described in "Setting Up Mail Services (Task Map)" on page 25. For other task information, refer to "Task Map for Mail Services" on page 21.

Local Mail Only

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

FIGURE 2–1 Local Mail Configuration



To set up this kind of mail configuration, you need the following.

- The default `/etc/mail/sendmail.cf` file, which requires no editing, on each mail client system.

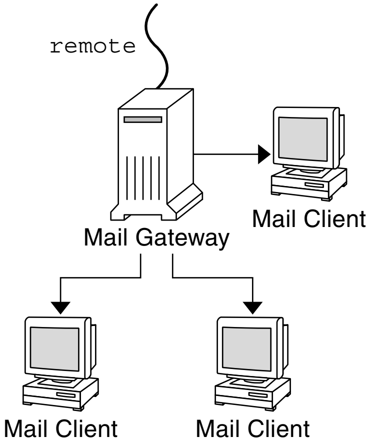
- A server that is designated as the mail host. If you are running NIS, you can make this designation by adding `mailhost.`*domain-name* to the `/etc/hosts` file on the mail host. If you are running another name service, such as DNS or LDAP, you must provide additional information in the `/etc/hosts` file. See "How to Set Up a Mail Host" on page 29.
- If you are using a name service other than NIS, you need matching `/etc/mail/aliases` files on any system that has a local mailbox.
- Enough space in `/var/mail` on each mail client system to hold the mailboxes.

For task information about setting up your mail service, refer to "Setting Up Mail Services" on page 25. If you are looking for a particular procedure that is related to setting up your mail service, refer to "Setting Up Mail Services (Task Map)" on page 25.

Local Mail and a Remote Connection

The most common mail configuration in a small network is shown in Figure 2–2. One system includes the mail server, the mail host, and the mail gateway that provides the remote connection. Mail is distributed by using the `/etc/mail/aliases` files on the mail gateway. No name service is required.

FIGURE 2–2 Local Mail Configuration With a UUCP Connection



In this configuration, you can assume that the mail clients mount their mail files from `/var/mail` on the mail host. To set up this kind of mail configuration, you need the following.

- The default `/etc/mail/sendmail.cf` file on each mail client system. This file does not require any editing.

- A server that is designated as the mail host. If you are running NIS, you can make this designation by adding `mailhost.`*domain-name* to the `/etc/hosts` file on the mail host. If you are running another name service, such as DNS or LDAP, you must provide additional information in the `/etc/hosts` file. See "How to Set Up a Mail Host" on page 29.
- If you are using a name service other than NIS, you need matching `/etc/mail/aliases` files on any system that has a local mailbox.
- Enough space in `/var/mail` on the mail server to hold the client mailboxes.

For task information about setting up your mail service, refer to "Setting Up Mail Services" on page 25. If you are looking for a particular procedure that is related to setting up your mail service, refer to "Setting Up Mail Services (Task Map)" on page 25.

Setting Up Mail Services (Task Map)

The following table describes the procedures for setting up mail services.

Task	Description	For Instructions
Setting up a mail server	Steps to enable a server to route mail	"How to Set Up a Mail Server" on page 26
Setting up a mail client	Steps to enable a user to receive mail	"How to Set Up a Mail Client" on page 27
Setting up a mail host	Steps to establish a mail host that can resolve email addresses	"How to Set Up a Mail Host" on page 29
Setting up a mail gateway	Steps to manage communication with networks outside your domain	"How to Set Up a Mail Gateway" on page 31
Using DNS with `sendmail`	Steps to enable DNS host lookups	"How to Use DNS With `sendmail`" on page 32

Setting Up Mail Services

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

Mail requires two types of configurations for local mail. Refer to Figure 2–1 in "Local Mail Only" on page 23 for a representation of these configurations. Mail requires two more configurations for communication with networks outside your domain. Refer to Figure 1–1 in "Overview of the Hardware Components" on page 18 or Figure 2–2 in "Local Mail and a Remote Connection" on page 24 for a representation of these configurations. You can combine these configurations on the same system or provide these configurations on separate systems. For example, if your mail host and mail server functions are on the same system, follow the directions in this section for setting up that system as a mail host. Then, follow the directions in this section for setting up the same system as a mail server.

Note – The following procedures for setting up a mail server and mail client apply when mailboxes are NFS mounted. However, mailboxes typically are maintained in locally mounted `/var/mail` directories, which eliminates the need for the following procedures.

▼ How to Set Up a Mail Server

No special steps are required to set up a mail server that is only serving mail for local users. The user must have an entry in the password file or in the namespace. Also, for mail to be delivered, the user should have a local home directory for checking the `~/.forward` file. For this reason, home directory servers are often set up as the mail server. "Hardware Components" on page 74 in Chapter 3, "Mail Services (Reference)," provides more information about the mail server.

The mail server can route mail for many mail clients. This type of mail server must have adequate spooling space for client mailboxes.

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

For clients to access their mailboxes, the `/var/mail` directory should be available for remote mounting. Alternately, a service such as Post Office Protocol (POP) or Internet Message Access Protocol (IMAP) should be available from the server. The following task shows you how to set up a mail server by using the `/var/mail` directory. To provide configuration guidelines for POP or IMAP is beyond the scope of this document.

For the following task, ensure that the `/etc/dfs/dfstab` file shows that the `/var/mail` directory is exported.

1. **Become an administrator.**

   For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2. **Stop `sendmail`.**

   ```
   # svcadm disable -t network/smtp:sendmail
   ```

3. **Check if the `/var/mail` directory is available for remote access.**

   ```
   # share
   ```

   If the `/var/mail` directory is listed, proceed to step 5.

If the /var/mail directory is not listed or if no list appears, continue with the appropriate substep.

a. **(Optional) If no list appears, start NFS services.**

   Follow the procedure, "How to Set Up Automatic File-System Sharing" in *Managing Network File Systems in Oracle Solaris 11.1*, to use the /var/mail directory to start NFS services.

b. **(Optional) If the /var/mail directory is not included in the list, add the directory to /etc/dfs/dfstab.**

   Add the following command line to the /etc/dfs/dfstab file.

   ```
   share -F nfs -o rw /var/mail
   ```

4 **Make the file system available for mounting.**

```
# shareall
```

5 **Ensure that your name service has been started.**

a. **(Optional) If you are running NIS, use this command.**

   ```
   # ypwhich
   ```

   For more information, refer to the ypwhich(1) man page.

b. **(Optional) If you are running DNS, use this command.**

   ```
   # nslookup hostname
   ```

   hostname Use your host name.

   For more information, refer to the nslookup(1M) man page.

c. **(Optional) If you are running LDAP, use this command.**

   ```
   # ldaplist
   ```

   For more information, refer to the ldaplist(1) man page.

6 **Restart sendmail.**

```
# svcadm enable network/smtp:sendmail
```

▼ How to Set Up a Mail Client

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

Note – You can also perform the task of setting up a mail client by using a service such as Post Office Protocol (POP) or Internet Message Access Protocol (IMAP). However, to provide configuration guidelines for POP or IMAP is beyond the scope of this document.

---

1. **Become an administrator on the mail client's system.**

   For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2. **Stop `sendmail`.**

   ```
   # svcadm disable -t network/smtp:sendmail
   ```

3. **Ensure that a `/var/mail` mount point exists on the mail client's system.**

   The mount point should have been created during the installation process. You can use `ls` to ensure that the file system exists. The following example shows the response that you receive if the file system has not been created.

   ```
   # ls -l /var/mail
   /var/mail not found
   ```

4. **Ensure that no files are in the `/var/mail` directory.**

   If mail files do exist in this directory, you should move them so that they are not covered when the `/var/mail` directory is mounted from the server.

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

   You can mount the mail directory automatically or at boot time.

   a. **(Optional) Mount `/var/mail` automatically.**

   Add an entry such as the following to the `/etc/auto_direct` file.

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

   server Use the assigned server name.

   b. **(Optional) Mount `/var/mail` at boot time.**

   Add the following entry to the `/etc/vfstab` file. This entry permits the `/var/mail` directory on the mail server that is specified to mount the local `/var/mail` directory.

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

   The client's mailbox is automatically mounted whenever the system is rebooted. If you are not rebooting the system, type the following command to mount the client mailbox.

   ```
   # mountall
   ```

Caution – For mailbox locking and mailbox access to work properly, you must include the `actimeo=0` option when mounting mail from an NFS server.

6 **Update `/etc/hosts`.**

Edit the `/etc/hosts` file and add an entry for the mail server. This step is not required if you are using a name service.

```
# cat /etc/hosts
#
# Internet host table
#
..
IP-address      mailhost  mailhost  mailhost.example.com
```

IP-address	Use the assigned IP addresses.
example.com	Use the assigned domain.
mailhost	Use the assigned mailhost.

For more information, refer to the `hosts(4)` man page.

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

Refer to "Administering Mail Alias Files (Task Map)" on page 42 for a task map about administering mail alias files. Note that the `mail.local` program automatically creates mailboxes in the `/var/mail` directory the first time a message is delivered. You do not need to create individual mailboxes for your mail clients.

8 **Restart `sendmail`.**

```
# svcadm enable network/smtp:sendmail
```

▼ How to Set Up a Mail Host

A mail host resolves email addresses and reroutes mail within your domain. A good candidate for a mail host is a system that provides your network with a remote connection or connects your network to a parent domain. The following procedure shows you how to set up a mail host.

1 **Become an administrator on the mail host system.**

For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2 **Stop `sendmail`.**

```
# svcadm disable -t network/smtp:sendmail
```

3 Verify the host-name configuration.

Run the `check-hostname` script to verify that `sendmail` can identify the fully qualified host name for this server.

```
% /usr/sbin/check-hostname
hostname phoenix OK: fully qualified as phoenix.example.com
```

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

4 Update the `/etc/hosts` file.

Choose the step that is appropriate for you.

a. (Optional) If you are using NIS, edit the `/etc/hosts` file on the system that is to be the new mail host.

Add the word `mailhost` and `mailhost.`*domain* after the IP address and system name of the mail host system.

IP-address mailhost `mailhost mailhost.`*domain* `loghost`

IP-address Use the assigned IP address.

mailhost Use the system name of the mail host system.

domain Use the expanded domain name.

The system is now designated as a mail host. The *domain* should be identical to the string that is given as the subdomain name in the output of the following command.

```
% /usr/lib/sendmail -bt -d0 </dev/null
Version 8.13.1+Sun
 Compiled with: LDAPMAP MAP_REGEX LOG MATCHGECOS MIME7TO8 MIME8TO7
                NAMED_BIND NDBM NETINET NETINET6 NETUNIX NEWDB NIS
                NISPLUS QUEUE SCANF SMTP USERDB XDEBUG

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

See the following example of how the `hosts` file should look after these changes.

```
# cat /etc/hosts
#
# Internet host table
#
172.31.255.255   localhost
192.168.255.255  phoenix mailhost mailhost.example.com loghost
```

b. **(Optional) If you are not using NIS, edit the `/etc/hosts` file on each system in the network.**

Create the following entry.

```
IP-address mailhost mailhost mailhost.domain loghost
```

5 **Restart `sendmail`.**

```
# svcadm enable network/smtp:sendmail
```

6 **Test your mail configuration.**

See "How to Test the Mail Configuration" on page 55 for instructions.

Note – For further information about mail hosts, refer to "Hardware Components" on page 74 in Chapter 3, "Mail Services (Reference)."

▼ How to Set Up a Mail Gateway

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

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

1 **Become an administrator on the mail gateway system.**

For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2 **Stop `sendmail`.**

```
# svcadm disable -t network/smtp:sendmail
```

3 **Verify the host-name configuration.**

Run the `check-hostname` script to verify that `sendmail` can identify the fully qualified host name for this server.

```
# /usr/sbin/check-hostname
hostname phoenix OK: fully qualified as phoenix.example.com
```

If this script is not successful in identifying the fully qualified host name, you need to add the fully qualified host name as the first alias for the host in `/etc/hosts`. If you need help with this step, refer to Step 4 of "How to Set Up a Mail Host" on page 29.

4 **Ensure that your name service has been started.**

a. **(Optional) If you are running NIS, use this command.**

```
# ypwhich
```

For more information, refer to the ypwhich(1) man page.

b. **(Optional) If you are running DNS, use this command.**

```
# nslookup hostname
```

hostname Use your host name.

For more information, refer to the nslookup(1M) man page.

c. **(Optional) If you are running LDAP, use this command.**

```
# ldaplist
```

For more information, refer to the ldaplist(1) man page.

5 **Restart `sendmail`.**

```
# svcadm enable network/smtp:sendmail
```

6 **Test your mail configuration.**

See "How to Test the Mail Configuration" on page 55 for instructions.

Note – For more information about the mail gateway, refer to "Hardware Components" on page 74 in Chapter 3, "Mail Services (Reference)."

▼ How to Use DNS With `sendmail`

The DNS name service does not support aliases for individuals. This name service does support aliases for hosts or domains that use Mail Exchanger (MX) records and `CNAME` records. You can specify host names, domain names, or both names in the DNS database. For more information about `sendmail` and DNS, see "Interactions of `sendmail` With Name Services" on page 93 in Chapter 3, "Mail Services (Reference)," or see the *Oracle Solaris Administration: Naming and Directory Services*.

1 **Become an administrator.**

For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

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

Use `nslookup` to ensure that an entry exists for `mailhost` and `mailhost.`*domain* in the DNS database. For more information, refer to the `nslookup(1M)` man page.

Changing the sendmail Configuration (Task Map)

Task	Description	For Instructions
Building a `sendmail` configuration file	Use this procedure to modify your `sendmail.cf` file. An example of how to enable domain masquerading is included.	"How to Build a New `sendmail.cf` File" on page 34
Setting up a virtual host	Steps to configure sendmail so that mail is accepted for more than one domain.	"Setting Up a Virtual Host" on page 35
Setting up automatic rebuilding of the sendmail configuration file	Use this procedure to modify the sendmail service so that the `sendmail.cf` and `submit.mc` configuration files are automatically rebuilt after an upgrade.	"How to Automatically Rebuild a Configuration File" on page 35
Running `sendmail` in the open mode.	Use this procedure to modify the sendmail service properties to enable the open mode.	"How to Use `sendmail` in the Open Mode" on page 36
Setting SMTP to use Transport Layer Security (TLS)	Use this procedure to enable SMTP to have secure connections with TLS.	"How to Set SMTP to Use TLS" on page 37
Managing mail delivery with an alternate configuration	Use this procedure to prevent mail delivery problems that can occur if the master daemon is disabled.	"How to Manage Mail Delivery by Using an Alternate Configuration of `sendmail.cf`" on page 41

Changing the sendmail Configuration

"How to Build a New `sendmail.cf` File" on page 34 shows you how to build the configuration file. Although you can still use older versions of `sendmail.cf` files, the best practice is to use the new format.

For more details, refer to the following.

- `/etc/mail/cf/README` provides a complete description of the configuration process.
- `http://www.sendmail.org` provides online information about `sendmail` configuration.
- "Versions of the Configuration File" on page 66 and "`sendmail` Configuration File" on page 87, in Chapter 3, "Mail Services (Reference)," provide some guidance.
- "Additional and Revised `m4` Configuration Macros From Version 8.12 of `sendmail`" on page 111 is also helpful.

▼ How to Build a New sendmail.cf File

The following procedure shows you how to build a new configuration file.

Note – /usr/lib/mail/cf/main-v7sun.mc is now /etc/mail/cf/cf/sendmail.mc.

---

1 **Become an administrator.**

For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2 **Stop sendmail.**

```
# svcadm disable -t network/smtp:sendmail
```

3 **Make a copy of the configuration files that you are changing.**

```
# cd /etc/mail/cf/cf
# cp sendmail.mc myhost.mc
```

myhost Select a new name for your .mc file.

4 **Edit the new configuration files (for example, *myhost*.mc), as necessary.**

For example, add the following command line to enable domain masquerading.

```
# cat myhost.mc
..
MASQUERADE_AS(‘host.domain')
```

host.domain Use the desired host name and domain name.

In this example, MASQUERADE_AS causes sent mail to be labeled as originating from *host.domain*, rather than $j.

5 **Build the configuration file by using m4.**

```
# make myhost.cf
```

6 **Test the new configuration file by using the -C option to specify the new file.**

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

While this command displays messages, it sends a message to testaddr. Only outgoing mail can be tested without restarting the sendmail service on the system. For systems that are not handling mail yet, use the full testing procedure in "How to Test the Mail Configuration" on page 55.

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

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

8 **Restart the sendmail service.**

```
# svcadm enable network/smtp:sendmail
```

Setting Up a Virtual Host

If you need to assign more than one IP address to a host, see this Web site: http://www.sendmail.org/tips/virtualHosting. This site provides complete instructions about how to use sendmail to set up a virtual host. However, in the "Sendmail Configuration" section, do not perform step 3b, as shown in the following.

```
# cd sendmail-VERSION/cf/cf
# ./Build mailserver.cf
# cp mailserver.cf /etc/mail/sendmail.cf
```

Instead, for the Oracle Solaris operating system, perform the following steps.

```
# cd /etc/mail/cf/cf
# make mailserver.cf
# cp mailserver.cf /etc/mail/sendmail.cf
```

mailserver Use the name of the .cf file.

"Changing the sendmail Configuration" on page 33 outlines the same three steps as part of the build process.

After you have generated your /etc/mail/sendmail.cf file, you can continue with the next steps to create a virtual user table.

▼ How to Automatically Rebuild a Configuration File

If you have built your own copy of sendmail.cf or submit.cf, the configuration file is not replaced during the upgrade process. The following procedure shows how to configure the sendmail service properties so that the sendmail.cf file is automatically rebuilt for you. For instructions on how to automatically build the submit.cf configuration file, see Example 2–1. You may combine these procedures if you need to build both files.

1 **Become an administrator.**

For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2 **Set the sendmail properties.**

```
# svccfg -s sendmail
svc:/network/smtp:sendmail> setprop config/path_to_sendmail_mc=/etc/mail/cf/cf/myhost.mc
svc:/network/smtp:sendmail> quit
```

3 **Refresh and restart the sendmail service.**

The first command pushes the changes into the running snapshot. The second command restarts the sendmail service using the new options.

```
# svcadm refresh svc:/network/smtp:sendmail
# svcadm restart svc:/network/smtp:sendmail
```

Example 2–1 Establishing Automatic Rebuilding of submit.cf

This procedure configures the sendmail service, such that the submit.mc configuration file is rebuilt automatically.

```
# svccfg -s sendmail-client:default
svc:/network/smtp:sendmail> setprop config/path_to_submit_mc=/etc/mail/cf/cf/submit-myhost.mc
svc:/network/smtp:sendmail> exit
# svcadm refresh svc:/network/sendmail-client
# svcadm restart svc:/network/sendmail-client
```

▼ How to Use sendmail in the Open Mode

The sendmail service has been changed so that it would run in local–only mode by default. The local-only mode means that only mail from the local host is accepted. Messages from any other systems are rejected. Earlier releases were configured to accept incoming mail from all remote systems, which is known as the open mode. To use the open mode, use the following procedure.

Caution – Running sendmail in the local–only mode is much more secure than running in the open mode. Make sure that you are aware of the potential security risks if you follow this procedure.

1 **Become an administrator.**

For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2 **Set the sendmail properties.**

```
# svccfg -s sendmail
svc:/network/smtp:sendmail> setprop config/local_only = false
svc:/network/smtp:sendmail> quit
```

3 **Refresh and restart the sendmail service.**

```
# svcadm refresh svc:/network/smtp:sendmail
# svcadm restart svc:/network/smtp:sendmail
```

▼ How to Set SMTP to Use TLS

SMTP can use Transport Layer Security (TLS) in version 8.13 of `sendmail`. This service to SMTP servers and clients provides private, authenticated communications over the Internet, as well as protection from eavesdroppers and attackers. Note that this service is not enabled by default.

The following procedure uses sample data to show you how to set up the certificates that enable `sendmail` to use TLS. For more information, see "Support for Running SMTP With TLS in Version 8.13 of `sendmail`" on page 97.

1 **Become an administrator.**

For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2 **Stop `sendmail`.**

```
# svcadm disable -t network/smtp:sendmail
```

3 **Set up the certificates that enable `sendmail` to use TLS.**

a. **Complete the following:**

```
# cd /etc/mail
# mkdir -p certs/CA
# cd certs/CA
# mkdir certs crl newcerts private
# echo "01" > serial
# cp /dev/null index.txt
# cp /etc/openssl/openssl.cnf .
```

b. **Use your preferred text editor to change the `dir` value in the `openssl.cnf` file from `/etc/openssl` to `/etc/mail/certs/CA`.**

c. **Use the `openssl` command-line tool to implement TLS.**

Note that the following command line generates interactive text.

```
# openssl req -new -x509 -keyout private/cakey.pem -out cacert.pem -days 365 \
-config openssl.cnf
Generating a 1024 bit RSA private key
.....................................++++++
.....................................++++++
writing new private key to 'private/cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) []:US
State or Province Name (full name) []:California
Locality Name (eg, city) []:Menlo Park
Organization Name (eg, company) [Unconfigured OpenSSL Installation]:Oracle
Organizational Unit Name (eg, section) []:Solaris
Common Name (eg, YOUR name) []:somehost.somedomain.example.com
Email Address []:someuser@example.com
```

`req`
This command creates and processes certificate requests.

`-new`
This `req` option generates a new certificate request.

`-x509`
This `req` option creates a self-signed certificate.

`-keyout private/cakey.pem`
This `req` option enables you to assign `private/cakey.pem` as the file name for your newly created private key.

`-out cacert.pem`
This `req` option enables you to assign `cacert.pem` as your output file.

`-days 365`
This `req` option enables you to certify the certificate for `365` days. The default value is `30`.

`-config openssl.cnf`
This `req` option enables you to specify `openssl.cnf` as the configuration file.

Note that this command requires that you provide the following:

- `Country Name`, such as `US`.
- `State or Province Name`, such as `California`.
- `Locality Name`, such as `Menlo Park`.
- `Organization Name`, such as `Oracle`.
- `Organizational Unit Name`, such as `Solaris`.
- `Common Name`, which is the machine's fully qualified host name. For more information, see the `check-hostname`(1M) man page.
- `Email Address`, such as `someuser@example.com`.

4 (Optional) If you need a new secure connection, make a new certificate and sign the new certificate with the certificate authority.

a. Make a new certificate.

```
# cd /etc/mail/certs/CA
# openssl req -nodes -new -x509 -keyout newreq.pem -out newreq.pem -days 365 \
-config openssl.cnf
Generating a 1024 bit RSA private key
..............++++++
..............++++++
writing new private key to 'newreq.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) []:US
State or Province Name (full name) []:California
Locality Name (eg, city) []:Menlo Park
Organization Name (eg, company) [Unconfigured OpenSSL Installation]:Oracle
Organizational Unit Name (eg, section) []:Solaris
Common Name (eg, YOUR name) []:somehost.somedomain.example.com
Email Address []:someuser@example.com
```

This command requires that you provide the same information that you provided in step 3c.

Note that in this example, the certificate and private key are in the file newreq.pem.

b. **Sign the new certificate with the certificate authority.**

```
# cd /etc/mail/certs/CA
# openssl x509 -x509toreq -in newreq.pem -signkey newreq.pem -out tmp.pem
Getting request Private Key
Generating certificate request
# openssl ca -config openssl.cnf -policy policy_anything -out newcert.pem -infiles tmp.pem
Using configuration from openssl.cnf
Enter pass phrase for /etc/mail/certs/CA/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 1 (0x1)
        Validity
            Not Before: Jun 23 18:44:38 2005 GMT
            Not After : Jun 23 18:44:38 2006 GMT
        Subject:
            countryName               = US
            stateOrProvinceName       = California
            localityName              = Menlo Park
            organizationName          = Oracle
            organizationalUnitName    = Solaris
            commonName                = somehost.somedomain.example.com
            emailAddress              = someuser@example.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                93:D4:1F:C3:36:50:C5:97:D7:5E:01:E4:E3:4B:5D:0B:1F:96:9C:E2
            X509v3 Authority Key Identifier:
                keyid:99:47:F7:17:CF:52:2A:74:A2:C0:13:38:20:6B:F1:B3:89:84:CC:68
                DirName:/C=US/ST=California/L=Menlo Park/O=Oracle/OU=Solaris/\
                CN=someuser@example.com/emailAddress=someuser@example.com
                serial:00
```

```
Certificate is to be certified until Jun 23 18:44:38 2006 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
# rm -f tmp.pem
```

In this example the file newreq.pem contains the unsigned certificate and private key. The file newcert.pem contains the signed certificate.

x509 utility Displays certificate information, converts certificates to various forms, and signs certificate requests

ca application Used to sign certificate requests in a variety of forms and to generate CRLs (certificate revocation lists)

5 **Enable sendmail to use the certificates by adding the following lines to your .mc file.**

```
define(‘confCACERT_PATH’, ‘/etc/mail/certs’)dnl
define(‘confCACERT’, ‘/etc/mail/certs/CAcert.pem’)dnl
define(‘confSERVER_CERT’, ‘/etc/mail/certs/MYcert.pem’)dnl
define(‘confSERVER_KEY’, ‘/etc/mail/certs/MYkey.pem’)dnl
define(‘confCLIENT_CERT’, ‘/etc/mail/certs/MYcert.pem’)dnl
define(‘confCLIENT_KEY’, ‘/etc/mail/certs/MYkey.pem’)dnl
```

For more information, see “Configuration File Options for Running SMTP With TLS” on page 98.

6 **Rebuild and install your sendmail.cf file in your /etc/mail directory.**

For detailed instructions, see “Changing the sendmail Configuration” on page 33.

7 **Create symbolic links from the files you created with openssl to the files you defined in your .mc file.**

```
# cd /etc/mail/certs
# ln -s CA/cacert.pem CAcert.pem
# ln -s CA/newcert.pem MYcert.pem
# ln -s CA/newreq.pem MYkey.pem
```

8 **For added security, deny read permission to group and others for MYkey.pem.**

```
# chmod go-r MYkey.pem
```

9 **Use a symbolic link to install CA certs in the directory assigned to confCACERT_PATH.**

```
# C=CAcert.pem
# ln -s $C ‘openssl x509 -noout -hash < $C‘.0
```

10 For secure mail with other hosts, install their host certificates.

a. Copy the file defined by the other host's `confCACERT` option to `/etc/mail/certs/`*host.domain*`.cert.pem`.

Replace *host.domain* with the other host's fully qualified host name.

b. Use a symbolic link to install CA certs in the directory assigned to `confCACERT_PATH`.

```
# C=host.domain.cert.pem
# ln -s $C ‘openssl x509 -noout -hash < $C‘.0
```

Replace *host.domain* with the other host's fully qualified host name.

11 Restart `sendmail`.

```
# svcadm enable network/smtp:sendmail
```

Example 2–2 Received: Mail Header

The following is an example of a `Received:` header for secure mail with TLS.

```
Received: from his.example.com ([IPv6:2001:db8:3c4d:15::1a2f:1a2b])
        by her.example.com (8.13.4+Sun/8.13.4) with ESMTP id j2TNUB8i242496
        (version=TLSv1/SSLv3 cipher=DHE-RSA-AES256-SHA bits=256 verify=OK)
        for <janepc@her.example.com>; Tue, 29 Mar 2005 15:30:11 -0800 (PST)
Received: from her.example.com (her.city.example.com [192.168.0.0])
        by his.example.com (8.13.4+Sun/8.13.4) with ESMTP id j2TNU7cl571102
        version=TLSv1/SSLv3 cipher=DHE-RSA-AES256-SHA bits=256 verify=OK)
        for <janepc@her.example.com>; Tue, 29 Mar 2005 15:30:07 -0800 (PST)
```

Note that the value for `verify` is `OK`, which means that the authentication was successful. For more information, see “Macros for Running SMTP With TLS” on page 100.

See Also The following OpenSSL man pages:

- openssl(1) (`http://www.openssl.org/docs/apps/openssl.html`).
- req(1) (`http://www.openssl.org/docs/apps/req.html`).
- x509(1) (`http://www.openssl.org/docs/apps/x509.html`).
- ca(1) (`http://www.openssl.org/docs/apps/ca.html`).

▼ How to Manage Mail Delivery by Using an Alternate Configuration of `sendmail.cf`

To facilitate the transport of inbound mail and outbound mail, the new default configuration of `sendmail` uses a daemon and a client queue runner. The client queue runner must be able to submit mail to the daemon on the local SMTP port. If the daemon is not listening on the SMTP port, the mail remains in the queue. To avoid this problem, perform the following task. For more information about the daemon and client queue runner and to understand why you might have to use this alternate configuration, refer to "`submit.cf` Configuration File From Version 8.12 of `sendmail`" on page 106.

This procedure ensures that your daemon runs only to accept connections from the local host.

1. **Become an administrator.**

   For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2. **Stop `sendmail` client service.**

   ```
   # svcadm disable -t sendmail-client
   ```

3. **Make a copy of the configuration file that you are changing.**

   ```
   # cd /etc/mail/cf/cf
   # cp submit.mc submit-myhost.mc
   ```

   myhost Select a new name for your `.mc` file.

4. **Edit the new configuration file (for example, `submit-`*myhost*`.mc`)**

   Change the listening host IP address to the `msp` definition.

   ```
   # grep msp submit-myhost.mc
   FEATURE('msp', '[#.#.#.#]')dnl
   ```

5. **Build the configuration file by using `m4`.**

   ```
   # make submit-myhost.cf
   ```

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

   ```
   # cp /etc/mail/submit.cf /etc/mail/submit.cf.save
   # cp submit-myhost.cf /etc/mail/submit.cf
   ```

7. **Restart the `sendmail` client service.**

   ```
   # svcadm enable sendmail-client
   ```

Administering Mail Alias Files (Task Map)

The following table describes the procedures for administering mail alias files. For more information about this topic, refer to "Mail Alias Files" on page 88 in Chapter 3, "Mail Services (Reference)."

Task	Description	For Instructions
Setting up an NIS mail.aliases map	If your name service is NIS, follow these instructions to facilitate aliasing with a mail.aliases map.	"How to Set Up an NIS mail.aliases Map" on page 43
Setting up a local mail alias file	If you are not using a name service (such as NIS), follow these instructions to facilitate aliasing with the /etc/mail/aliases file.	"How to Set Up a Local Mail Alias File" on page 44
Creating a keyed map file	Use these steps to facilitate aliasing with a keyed map file.	"How to Create a Keyed Map File" on page 46
Setting up the postmaster alias	Use the procedures in this section to manage the postmaster alias. You must have this alias.	"Managing the postmaster Alias" on page 46

Administering Mail Alias Files

Mail aliases must be unique within the domain. This section provides the procedures for administering mail alias files.

In addition, you can create database files for the local mail host by using makemap. Refer to the makemap(1M) man page. The use of these database files does not provide all of the advantages of using a name service such as NIS. However, you should be able to retrieve the data from these local database files faster because no network lookups are involved. For more information, refer to "Interactions of sendmail With Name Services" on page 93 and "Mail Alias Files" on page 88 in Chapter 3, "Mail Services (Reference)."

▼ How to Set Up an NIS mail.aliases Map

Use the following procedure to facilitate aliasing with an NIS mail.aliases map.

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

2 **Become an administrator on the NIS master server.**

For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

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

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

```
# cat /etc/mail/aliases
..
alias:expanded-alias
```

alias	Use the short alias name.
expanded-alias	Use the expanded alias name (`user@host.domain.com`).

b. **Ensure that you have a `Postmaster: root` entry.**

```
# cat /etc/mail/aliases
..
Postmaster: root
```

c. **Add an alias for `root`. Use the mail address of the person who is designated as the postmaster.**

```
# cat /etc/mail/aliases
..
root: user@host.domain.com
```

user@host.domain.com	Use the assigned address of the designated postmaster.

4 **Ensure that the NIS master server is running a name service to resolve the host names on each mail server.**

5 **Change to the `/var/yp` directory.**

```
# cd /var/yp
```

6 **Apply the `make` command.**

```
# make
```

The changes in the `/etc/hosts` and `/etc/mail/aliases` files are propagated to NIS slave systems. The changes are active in only a few minutes, at most.

▼ How to Set Up a Local Mail Alias File

Use the following procedure to resolve aliases with a local mail alias file.

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

2 **Become an administrator on the mail server.**

For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

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

a. **Add an entry for each user.**

```
user1: user2@host.domain
```

user1	Use the new alias name.

user2@host.domain Use the actual address for the new alias.

b. **Ensure that you have a `Postmaster: root` entry.**

```
# cat /etc/mail/aliases
..
Postmaster: root
```

c. **Add an alias for `root`. Use the mail address of the person who is designated as the postmaster.**

```
# cat /etc/mail/aliases
..
root: user@host.domain.com
```

user@host.domain.com Use the assigned address of the designated postmaster.

4 **Rebuild the alias database.**

```
# newaliases
```

The configuration of the `AliasFile` option in `/etc/mail/sendmail.cf` determines whether this command generates in binary form either the single file, `/etc/mail/aliases.db`, or the pair of files, `/etc/mail/aliases.dir` and `/etc/mail/aliases.pag`.

5 **Perform one of the following steps to copy the file or files that were generated.**

a. **(Optional) Copy the `/etc/mail/aliases`, the `/etc/mail/aliases.dir`, and the`/etc/mail/aliases.pag` files to each of the other systems.**

You can copy the three files by using the `rcp` or `rsync` commands. Refer to the `rcp`(1) man page or the `rsync`(1) man page for more information. Alternately, you can create a script for this purpose.

When you copy these files, you do not need to run the `newaliases` command on each of the other systems. However, remember that you must update all the `/etc/mail/aliases` files each time you add or remove a mail client.

b. **(Optional) Copy the `/etc/mail/aliases` and the `/etc/mail/aliases.db` files to each of the other systems.**

You can copy these files by using the `rcp` or `rsync` commands. Refer to the `rcp`(1) man page or the `rsync`(1) man page for more information. Alternately, you can create a script for this purpose.

When you copy these files, you do not need to run the `newaliases` command on each of the other systems. However, remember that you must update all the `/etc/mail/aliases` files each time you add or remove a mail client.

▼ How to Create a Keyed Map File

To create a keyed map file, follow these instructions.

1 **Become an administrator.**

For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2 **Create an input file.**

Entries can have the following syntax.

```
old-name@newdomain.com     new-name@newdomain.com
old-name@olddomain.com     error:nouser No such user here
@olddomain.com            %1@newdomain.com
```

old_name@newdomain.com Use the user name that was previously assigned with the domain that is newly assigned.

new_name@newdomain.com Use the address that is newly assigned.

old_name@olddomain.com Use the user name that was previously assigned with the domain that was previously assigned.

olddomain.com Use the domain that was previously assigned.

newdomain.com Use the domain that is newly assigned.

The first entry redirects mail to a new alias. The next entry creates a message when an incorrect alias is used. The last entry redirects all incoming mail from `olddomain` to `newdomain`.

3 **Create the database file.**

```
# /usr/sbin/makemap maptype newmap < newmap
```

maptype Select a database type, such as `dbm`, `btree`, or `hash`.

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

Managing the postmaster Alias

Every system must be able to send mail to a `postmaster` mailbox. You can create an NIS alias for `postmaster`, or you can create the alias in each local `/etc/mail/aliases` file. Refer to these procedures.

- "How to Create a `postmaster` Alias in Each Local `/etc/mail/aliases` File" on page 47
- "How to Create a Separate Mailbox for `postmaster`" on page 47

- "How to Add the `postmaster` Mailbox to the Aliases in the `/etc/mail/aliases` File" on page 48

How to Create a `postmaster` Alias in Each Local `/etc/mail/aliases` File

If you are creating the `postmaster` alias in each local `/etc/mail/aliases` file, follow these instructions.

1 **Become an administrator.**

For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2 **View the `/etc/mail/aliases` entry.**

```
# cat /etc/mail/aliases
# Following alias is required by the mail protocol, RFC 2821
# Set it to the address of a HUMAN who deals with this system’s
# mail problems.
Postmaster: root
```

3 **Edit each system's `/etc/mail/aliases` file.**

Change `root` to the mail address of the person who is designated as the postmaster.

`Postmaster:` *mail-address*

mail-address Use the assigned address for the person who is designated as the postmaster.

4 **(Optional) Create a separate mailbox for the postmaster.**

You can create a separate mailbox for the postmaster to keep postmaster mail separate from personal mail. If you create a separate mailbox, use the mailbox address instead of the postmaster's personal mail address when you edit the `/etc/mail/aliases` files. For details, refer to "How to Create a Separate Mailbox for `postmaster`" on page 47.

How to Create a Separate Mailbox for postmaster

If you are creating a separate mailbox for `postmaster`, follow these instructions.

1 **Become an administrator.**

For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2 **Create a user account for the person who is designated as `postmaster`. Put an asterisk (*) in the password field.**

For details about adding a user account, refer to "Setting Up and Managing User Accounts by Using the CLI (Task Map)" in *Managing User Accounts and User Environments in Oracle Solaris 11.1*.

3 **After mail has been delivered, enable the `mail` program to read and write to the mailbox name.**

```
# mail -f postmaster
```

postmaster	Use the assigned address.

▼ How to Add the postmaster Mailbox to the Aliases in the `/etc/mail/aliases` File

If you are adding a `postmaster` mailbox to the aliases in the `/etc/mail/aliases` file, follow these instructions.

1 **Become an administrator.**

For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2 **Add an alias for `root`. Use the mail address of the person who is designated as the postmaster.**

```
# cat /etc/mail/aliases
..
root: user@host.domain.com
```

user@host.domain.com	Use the assigned address of the person who is designated as postmaster.

3 **On the postmaster's local system, create an entry in the `/etc/mail/aliases` file that defines the name of the alias. `sysadmin` is an example. Also, include the path to the local mailbox.**

```
# cat /etc/mail/aliases
..
sysadmin: /usr/somewhere/somefile
```

sysadmin	Create a name for a new alias.
/usr/somewhere/somefile	Use the path to the local mailbox.

4 **Rebuild the alias database.**

```
# newaliases
```

Administering the Queue Directories (Task Map)

The following table describes the procedures for administering the mail queue.

Task	Description	For Instructions
Displaying the contents of the mail queue, /var/spool/mqueue	Use this procedure to see how many messages are in the queue and how fast the messages are being cleared from the queue.	"How to Display the Contents of the Mail Queue, /var/spool/mqueue" on page 49
Forcing mail queue processing for the mail queue, /var/spool/mqueue	Use this procedure to process messages to a system that previously was unable to receive messages.	"How to Force Mail Queue Processing in the Mail Queue, /var/spool/mqueue" on page 50
Running a subset of the mail queue, /var/spool/mqueue	Use this procedure to force a substring of an address, such as a host name, to be processed. Also, use this procedure to force a particular message out of the queue.	"How to Run a Subset of the Mail Queue, /var/spool/mqueue" on page 50
Moving the mail queue, /var/spool/mqueue	Use this procedure to move the mail queue.	"How to Move the Mail Queue, /var/spool/mqueue" on page 51
Running the old mail queue, /var/spool/omqueue	Use this procedure to run an old mail queue.	"How to Run the Old Mail Queue, /var/spool/omqueue" on page 51

Administering the Queue Directories

This section describes some helpful tasks for queue administration. For information about the client-only queue, refer to "submit.cf Configuration File From Version 8.12 of sendmail" on page 106. For other related information, you can refer to "Additional Queue Features From Version 8.12 of sendmail" on page 117.

Refer to the following:

- "How to Display the Contents of the Mail Queue, /var/spool/mqueue" on page 49
- "How to Force Mail Queue Processing in the Mail Queue, /var/spool/mqueue" on page 50
- "How to Run a Subset of the Mail Queue, /var/spool/mqueue" on page 50
- "How to Move the Mail Queue, /var/spool/mqueue" on page 51
- "How to Run the Old Mail Queue, /var/spool/omqueue" on page 51

▼ How to Display the Contents of the Mail Queue, /var/spool/mqueue

● **Show how many messages are in the queue and how fast they are being cleared from the queue.**

Type the following:

```
# /usr/bin/mailq | more
```

This command provides the following information.

- The queue IDs

- The size of the message
- The date that the message entered the queue
- The message status
- The sender and the recipients

Additionally, this command now checks for the authorization attribute, `solaris.admin.mail.mailq`. If the check is successful, the equivalent of specifying the `-bp` flag with `sendmail` is executed. If the check fails, an error message is printed. By default, this authorization attribute is enabled for all users. The authorization attribute can be disabled by modifying the user entry in `prof_attr`. For more information, refer to the man pages for `prof_attr`(4) and `mailq`(1).

▼ How to Force Mail Queue Processing in the Mail Queue, `/var/spool/mqueue`

Use this procedure, for example, to process messages to a system that was previously unable to receive messages.

1 **Become an administrator.**

For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2 **Force queue processing and display the progress of the jobs as the queue is cleared.**

```
# /usr/lib/sendmail -q -v
```

▼ How to Run a Subset of the Mail Queue, `/var/spool/mqueue`

Use this procedure, for example, to force a substring of an address, such as a host name, to be processed. Also, use this procedure to force a particular message from the queue.

1 **Become an administrator.**

For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2 **Run a subset of the mail queue at any time with `-qR`*string*.**

```
# /usr/lib/sendmail -qRstring
```

string Use a recipient's alias or a substring of *user@host.domain*, such as a host name.

Alternately, you can run a subset of the mail queue with `-qI`*nnnnn*.

```
# /usr/lib/sendmail -qInnnnn
```

nnnnn Use a queue ID.

▼ How to Move the Mail Queue, `/var/spool/mqueue`

If you are moving the mail queue, follow these instructions.

1 **Become an administrator on the mail host.**

For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2 **Kill the `sendmail` daemon.**

```
# svcadm disable network/smtp:sendmail
```

Now `sendmail` is no longer processing the queue directory.

3 **Change to the `/var/spool` directory.**

```
# cd /var/spool
```

4 **Move the directory, `mqueue`, and all its contents to the `omqueue` directory. Then create a new empty directory that is named `mqueue`.**

```
# mv mqueue omqueue; mkdir mqueue
```

5 **Set the permissions of the directory to read/write/execute by owner, and read/execute by group. Also, set the owner and group to `daemon`.**

```
# chmod 750 mqueue; chown root:bin mqueue
```

6 **Start `sendmail`.**

```
# svcadm enable network/smtp:sendmail
```

▼ How to Run the Old Mail Queue, `/var/spool/omqueue`

To run an old mail queue, follow these instructions.

1 **Become an administrator.**

For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2 **Run the old mail queue.**

```
# /usr/lib/sendmail -oQ/var/spool/omqueue -q
```

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

3 **Remove the empty directory.**

```
# rmdir /var/spool/omqueue
```

Administering .forward Files (Task Map)

The following table describes the procedures for administering .forward files. For more information, refer to ".forward Files" on page 90 in Chapter 3, "Mail Services (Reference)."

Task	Description	For Instructions
Disabling .forward files	Use this procedure if, for example, you want to prevent automated forwarding.	"How to Disable .forward Files" on page 52
Changing the .forward file search path	Use this procedure if, for example, you want to move all .forward files into a common directory.	"How to Change the .forward–File Search Path" on page 53
Creating and populating /etc/shells	Use this procedure to enable users to use the .forward file to forward mail to a program or to a file.	"How to Create and Populate /etc/shells" on page 54

Administering .forward Files

This section contains several procedures that are related to .forward file administration. Because these files can be edited by users, the files can cause problems. For more information, refer to ".forward Files" on page 90 in Chapter 3, "Mail Services (Reference)."

Refer to the following:

- "How to Disable .forward Files" on page 52
- "How to Change the .forward–File Search Path" on page 53
- "How to Create and Populate /etc/shells" on page 54

▼ How to Disable .forward Files

This procedure, which prevents automated forwarding, disables the .forward file for a particular host.

1 **Become an administrator.**

For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2 **Make a copy of /etc/mail/cf/domain/solaris-generic.m4 or your site-specific domain m4 file.**

```
# cd /etc/mail/cf/domain
# cp solaris-generic.m4 mydomain.m4
```

mydomain Use the file name of your choice.

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

```
define(‘confFORWARD_PATH’,‘’)dnl
```

If a value for confFORWARD_PATH already exists in the m4 file, replace the value with this null value.

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

If you need help with this step, refer to "How to Build a New sendmail.cf File" on page 34.

Note – When you edit the .mc file, remember to change DOMAIN(‘solaris-generic’) to DOMAIN(‘*mydomain*’).

▼ How to Change the .forward–File Search Path

If, for example, you want to put all .forward files in a common directory, follow these instructions.

1 **Become an administrator.**

For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2 **Make a copy of /etc/mail/cf/domain/solaris-generic.m4 or your site-specific domain m4 file.**

```
# cd /etc/mail/cf/domain
# cp solaris-generic.m4 mydomain.m4
```

mydomain Use the file name of your choice.

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

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

If a value for confFORWARD_PATH already exists in the m4 file, replace the value with this new value.

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

If you need help with this step, refer to "How to Build a New sendmail.cf File" on page 34.

Note – When you edit the `.mc` file, remember to change `DOMAIN('solaris-generic')` to `DOMAIN('`*mydomain*`')`.

▼ How to Create and Populate `/etc/shells`

This file is not included in the standard release. You must add the file if users are to be allowed to use `.forward` files to forward mail to a program or to a file. You can create the file manually by using `grep` to identify all of the shells that are listed in your password file. You can then type the shells into the file. However, the following procedure, which employs a script that can be downloaded, is easier to use.

1 **Download the script.**

`http://www.sendmail.org/vendor/sun/gen-etc-shells.html`

2 **Become an administrator.**

For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

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

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

This script uses the `getent` command to collect the names of shells that are included in the password file sources that are listed in the `svc:/system/name-service/switch` service.

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

With the editor of your choice, remove any shells that you are not including.

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

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

Troubleshooting Procedures and Tips for Mail Services (Task Map)

The following table describes troubleshooting procedures and tips for mail services.

Task	Description	For Instructions
Testing mail configuration	Steps for testing changes to the `sendmail` configuration file	"How to Test the Mail Configuration" on page 55
Checking mail aliases	A step to confirm that mail can or cannot be delivered to a specified recipient	"How to Check Mail Aliases" on page 56
Testing the rule sets	Steps for checking the input and returns of the `sendmail` rule sets	"How to Test the `sendmail` Rule Sets" on page 57
Verifying connections to other systems	Tips for verifying connections to other systems	"How to Verify Connections to Other Systems" on page 57
Logging messages by using the `syslogd` program	Tips for gathering error message information	"Logging Error Messages" on page 58
Checking other sources for diagnostic information	Tips for getting diagnostic information from other sources	"Other Sources for Mail Diagnostic Information" on page 59

Troubleshooting Procedures and Tips for Mail Services

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

▼ How to Test the Mail Configuration

To test the changes that you make to your configuration file, follow these instructions.

1. **Restart `sendmail` on any system that has a revised configuration file.**

```
# svcadm refresh network/smtp:sendmail
```

2. **Send test messages from each system.**

```
# /usr/lib/sendmail -v names </dev/null
```

names Specify a recipient's email address.

This command sends a null message to the specified recipient and displays the message activity on your monitor.

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

4. **(Optional) If you are connected to a network, send mail in three directions to someone on another system.**

   - From the main system to a client system

- From a client system to the main system
- From a client system to another client system

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

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

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

The `sendmail` program cannot detect whether the message is delivered because the program passes the message to UUCP for delivery.

8 **From different systems, send a message to `postmaster` and ensure that the message is delivered to your postmaster's mailbox.**

How to Check Mail Aliases

The following example shows you how to verify an alias.

```
% mconnect
connecting to host localhost (127.0.0.1), port 25
connection open
220 your.domain.com ESMTP Sendmail 8.13.6+Sun/8.13.6; Tue, 12 Sep 2004 13:34:13 -0800 (PST)
expn sandy
250 2.1.5 <sandy@phoenix.example.com>
quit
221 2.0.0 your.domain.com closing connection
%
```

In this example, the `mconnect` program opened a connection to a mail server on a local host and enabled you to test that connection. The program runs interactively, so you can issue various diagnostic commands. For a complete description, see the `mconnect(1)` man page. The entry, `expn sandy`, provided the expanded address, `sandy@phoenix.example.com`. Thus, you have verified that mail can be delivered when using the alias `sandy`.

Remember to avoid loops and inconsistent databases when both local and domain-wide aliases are used. Be especially careful to avoid the creation of alias loops when you move a user from one system to another system.

▼ How to Test the sendmail Rule Sets

To check the input and returns of the sendmail rule sets, follow these instructions.

1. **Change to address test mode.**

```
# /usr/lib/sendmail -bt
```

2. **Test a mail address.**

   Provide the following numbers and address at the last prompt (>).

```
> 3,0 mail-sraddress
```

mail-address Use the mail address that you are testing.

3. **End the session.**

   Press Control-d.

Example 2–3 Address Test Mode Output

The following is an example of the output from the address test mode.

```
% /usr/lib/sendmail -bt
ADDRESS TEST MODE (ruleset 3 NOT automatically invoked)
Enter <ruleset> <address>
> 3,0 sandy@phoenix
canonify           input: sandy @ phoenix
Canonify2          input: sandy < @ phoenix >
Canonify2        returns: sandy < @ phoenix . example . com . >
canonify         returns: sandy < @ phoenix . example . com . >
parse              input: sandy < @ phoenix . example . com . >
Parse0             input: sandy < @ phoenix . example . com . >
Parse0           returns: sandy < @ phoenix . example . com . >
ParseLocal         input: sandy < @ phoenix . example . com . >
ParseLocal       returns: sandy < @ phoenix . example . com . >
Parse1             input: sandy < @ phoenix . example . com . >
MailerToTriple     input: < mailhost . phoenix . example . com >
     sandy < @ phoenix . example . com . >
MailerToTriple   returns: $# relay $@ mailhost . phoenix . example . com
     $: sandy < @ phoenix . example . com . >
Parse1           returns: $# relay $@ mailhost . phoenix . example . com
     $: sandy < @ phoenix . example . com . >
parse            returns: $# relay $@ mailhost . phoenix . example . com
     $: sandy < @ phoenix . example . com . >
```

How to Verify Connections to Other Systems

The mconnect program opens a connection to a mail server on a host that you specify and enables you to test that connection. The program runs interactively, so you can issue various diagnostic commands. See the mconnect(1) man page for a complete description. The following example verifies that mail to the user name sandy is deliverable.

```
% mconnect phoenix

connecting to host phoenix (172.31.255.255), port 25
connection open
220 phoenix.example.com ESMTP Sendmail 8.13.1+Sun/8.13.1; Sat, 4 Sep 2004 3:52:56 -0700
expn sandy
250 2.1.5 <sandy@phoenix.example.com>
quit
```

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

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

Logging Error Messages

Your mail service logs most error messages by using the syslogd program. By default, the syslogd program sends these messages to a system that is called loghost, which is specified in the /etc/hosts file. You can define loghost to hold all logs for an entire NIS domain. If no loghost is specified, error messages from syslogd are not reported.

The /etc/syslog.conf file controls where the syslogd program forwards messages. You can change the default configuration by editing the /etc/syslog.conf file. You must restart the syslog daemon for any changes to become active. To gather information about mail, you can add the following selections to the file.

- mail.alert – Messages about conditions that should be fixed now
- mail.crit – Critical messages
- mail.warning – Warning messages
- mail.notice – Messages that are not errors, but might need attention
- mail.info – Informational messages
- mail.debug – Debugging messages

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

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

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

The log is arranged in a succession of levels. At the lowest level, only unusual occurrences are logged. At the highest level, even the most mundane and uninteresting events are recorded. As a convention, log levels under 10 are considered "useful." Log levels that are higher than 10 are usually used for debugging. See "Customizing System Message Logging" in *Troubleshooting Typical Issues in Oracle Solaris 11.1* for information about `loghost` and the `syslogd` program.

Other Sources for Mail Diagnostic Information

For other diagnostic information, check the following sources.

- Look at the `Received` lines in the header of the message. These lines trace the route that the message took as the message was relayed. Remember to consider time–zone differences.
- Look at the messages from `MAILER-DAEMON`. These messages typically report delivery problems.
- Check the system log that records delivery problems for your group of systems. The `sendmail` program always records its activities in the system log. You might want to modify the `crontab` file to run a shell script nightly. The script searches the log for `SYSERR` messages and mails any messages that it finds to the postmaster.
- Use the `mailstats` program to test mail types and determine the number of incoming messages and outgoing messages.

Resolving Error Messages

This section describes how you can resolve some `sendmail`–related error messages. You can also refer to `http://www.sendmail.org/faq`.

The following error messages contain two or more of the following types of information.

- **Cause:** What might have happened to cause the message
- **Description:** What the user was doing when the error message occurred
- **Solution:** What you can do to fix the problem or to continue with your work

`451 timeout waiting for input during` *source*

Cause: When `sendmail` reads from any source that might time out, such as an SMTP connection, the program sets a timer to the value of various `Timeout` options before reading begins. If the read is not completed before the timer expires, this message appears and reading stops. Usually, this situation occurs during `RCPT`. The mail message is then queued for later delivery.

Solution: If you see this message often, increase the value of various `Timeout` options in the `/etc/mail/sendmail.cf` file. If the timer is already set to a large number, look for hardware problems, such as poor network cabling or connections.

`550` *hostname*`... Host unknown`

Cause: This `sendmail` message indicates that the destination host machine, which is specified by the portion of the address after the at sign (@), was not found during domain name system (DNS) lookup.

Solution: Use the `nslookup` command to verify that the destination host exists in that domain or other domains, perhaps with a slightly different spelling. Otherwise, contact the intended recipient and ask for a proper address.

`550` *username*`... User unknown`

Cause: This `sendmail` message indicates that the intended recipient, who is specified by the portion of the address before the at sign (@), could not be located on the destination host machine.

Solution: Check the email address and try again, perhaps with a slightly different spelling. If this remedy does not work, contact the intended recipient and ask for a proper address.

`554` *hostname*`... Local configuration error`

Cause: This `sendmail` message usually indicates that the local host is trying to send mail to itself.

Solution: Check the value of the `$j` macro in the `/etc/mail/sendmail.cf` file to ensure that this value is a fully qualified domain name.

Description: When the sending system provides its host name to the receiving system in the SMTP `HELO` command, the receiving system compares its name to the sender's name. If these names are the same, the receiving system issues this error message and closes the connection. The name that is provided in the `HELO` command is the value of the `$j` macro.

For additional information, refer to `http://www.sendmail.org/faq/section4#4.5`.

`config error: mail loops back to myself.`

Cause: This error message occurs if you set up an MX record and make host *bar* the mail exchanger for domain *foo*. However, you fail to configure host *bar* to know that it is the mail exchanger for domain *foo*.

Also, another possibility is that both the sending system and the receiving system are identifying as the same domain.

Solution: For instructions, refer to `http://www.sendmail.org/faq/section4#4.5`.

`host name configuration error`

Description: This is an old `sendmail` message, which replaced `I refuse to talk to myself` and is now replaced by the `Local configuration error` message.

Solution: Follow the instructions that were provided for resolving this error message, `554` *hostname*`... Local configuration error`.

`user unknown`

Cause: When you try to send mail to a user, the error `Username... user unknown` is displayed. The user is on the same system.

Solution: Check for a typographical error in the entered email address. Otherwise, the user could be aliased to a nonexistent email address in `/etc/mail/aliases` or in the user's `.mailrc` file. Also, check for uppercase characters in the user name. Preferably, email addresses should not be case sensitive.

For additional information, refer to `http://www.sendmail.org/faq/section4#4.17`.

CHAPTER 3

Mail Services (Reference)

The sendmail program is a mail transport agent. The program uses a configuration file to provide aliasing and forwarding, automatic routing to network gateways, and flexible configuration. The Oracle Solaris OS supplies standard configuration files that most sites can use. Chapter 1, “Mail Services (Overview),” provides an introduction to the components of mail services and a description of a typical mail service configuration. Chapter 2, “Mail Services (Tasks),” explains how to set up and administer an electronic mail system. This chapter provides information about the following topics.

For details that are not covered in these chapters, see the following man pages:

- sendmail(1M)
- mail.local(1M)
- mailstats(1)
- makemap(1M)
- editmap(1M)

Oracle Solaris Version of sendmail

This section, which includes the following topics, describes some of the differences in the Oracle Solaris version of sendmail as compared to the generic Berkeley version.

- "Flags Used and Not Used to Compile sendmail" on page 64
- "MILTER, Mail Filter API for sendmail" on page 65
- "Alternative sendmail Commands" on page 66
- "Versions of the Configuration File" on page 66

Flags Used and Not Used to Compile sendmail

The following flags are used to compile sendmail. If your configuration requires other flags, you need to download the source and recompile the binary. You can find information about this process at http://www.sendmail.org.

TABLE 3–1 General sendmail Flags

Flag	Description
SOLARIS=21000	Support for the Solaris 10 release.
MILTER	Support for the Mail Filter API. In version 8.13 of sendmail, this flag is enabled by default. See "MILTER, Mail Filter API for sendmail" on page 65.
NETINET6	Support for IPv6. This flag has been moved from conf.h to Makefile.

TABLE 3–2 Maps and Database Types

Flag	Description
NDBM	Support for ndbm databases
NEWDB	Support for Berkeley DB databases
USERDB	Support for the user database
NIS	Support for nis databases
NISPLUS	Support for nisplus databases
LDAPMAP	Support for LDAP maps
MAP_REGEX	Support for regular expression maps

TABLE 3–3 OS Flags

Flag	Description
SUN_EXTENSIONS	Support for extensions that are included in sun_compat.o.
SUN_INIT_DOMAIN	For backward compatibility, support for the use of NIS domain names to fully qualify the local host name. For more information, look for vendor-specific information in http://www.sendmail.org.
SUN_SIMPLIFIED_LDAP	Support for a simplified LDAP API, which is specific to Sun. For more information, look for vendor-specific information in http://www.sendmail.org.
VENDOR_DEFAULT=VENDOR_SUN	Selects Sun as the default vendor.

The following table lists generic flags that are not used to compile the version of sendmail.

TABLE 3–4 Generic Flags Not Used in This Version of sendmail

Flag	Description
SASL	Simple Authentication and Security Layer (RFC 2554)
STARTTLS	Transaction Level Security (RFC 2487)

To see a list of the flags that are used to compile sendmail, use the following command.

```
% /usr/lib/sendmail -bt -d0.10 < /dev/null
```

Note – The preceding command does not list the flags that are specific to Sun.

MILTER, Mail Filter API for sendmail

MILTER, sendmail's Mail Filter API, enables third-party programs to access mail messages as they are being processed to filter meta-information and content. You do not need to build the filter and configure sendmail to use it. This API is enabled by default in version 8.13 of sendmail.

For more details, see the following:

- http://www.sendmail.org
- https://www.milter.org/

Alternative sendmail Commands

The Oracle Solaris release does not include all of the command synonyms that are provided in the generic release from sendmail.org. This table includes a complete list of the command aliases. The table also lists whether the commands are included in the Oracle Solaris release and how to generate the same behavior by using sendmail.

TABLE 3–5 Alternate sendmail Commands

Alternate Name	In This Release?	Options With sendmail
hoststat	No	sendmail -bh
mailq	Yes	sendmail -bp
newaliases	Yes	sendmail -bi
purgestat	No	sendmail -bH
smtpd	No	sendmail -bd

Versions of the Configuration File

sendmail includes a configuration option that enables you to define the version of the sendmail.cf file. This option enables older configuration files to be used with the current version of sendmail. You can set the version level to values between 0 and 10. You can also define the vendor. Either Berkeley or Sun is a valid vendor option. If a version level is specified but no vendor is defined, Sun is used as the default vendor setting. The following table lists some of the valid options.

TABLE 3–6 Version Values for the Configuration File

Field	Description
V7/Sun	Setting that was used for version 8.8 of sendmail.
V8/Sun	Setting that was used for version 8.9 of sendmail. This setting was included in the Solaris 8 release.
V9/Sun	Setting that was used for versions 8.10 and 8.11 of sendmail.
V10/Sun	Setting that is used for version 8.12, 8.13 and 8.14 versions of sendmail. Version 8.12 is the default for the Solaris 9 release. Starting in the Solaris 10 release, version 8.13 is the default. Version 8.14 is the default for the Oracle Solaris 11 release.

Note – You are urged not to use V1/Sun. For more information, refer to http://www.sendmail.org/vendor/sun/differences.html#4.

For task information, refer to "Changing the sendmail Configuration" on page 33 in Chapter 2, "Mail Services (Tasks)."

Software and Hardware Components of Mail Services

This section describes the software and hardware components of a mail system.

- "Software Components" on page 67
- "Hardware Components" on page 74

Software Components

Each mail service includes at least one of each of the following software components.

- "Mail User Agent" on page 67
- "Mail Transfer Agent" on page 67
- "Local Delivery Agent" on page 68

This section also describes these software components.

- "Mailers and sendmail" on page 68
- "Mail Addresses" on page 69
- "Mailbox Files" on page 71
- "Mail Aliases" on page 73

Mail User Agent

The *mail user agent* is the program that acts as the interface between the user and mail transfer agent. The `sendmail` program is a mail transfer agent. The Oracle Solaris operating system supplies the following mail user agents.

- `/usr/bin/mail`
- `/usr/bin/mailx`

Mail Transfer Agent

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

- Accepts messages from the mail user agent

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

Local Delivery Agent

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

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

"Changes From Version 8.12 of `sendmail`" on page 105 provides information on these related topics.

- "Additional Delivery Agent Flags From Version 8.12 of `sendmail`" on page 116
- "Additional Equates for Delivery Agents From Version 8.12 of `sendmail`" on page 116

Mailers and `sendmail`

Mailer is a `sendmail`-specific term. A *mailer* is used by `sendmail` to identify a specific instance of a customized local delivery agent or a customized mail transfer agent. You need to specify at least one mailer in your `sendmail.cf` file. For task information, refer to "Changing the `sendmail` Configuration" on page 33 in Chapter 2, "Mail Services (Tasks)." This section provides a brief description of two types of mailers.

- "Simple Mail Transfer Protocol (SMTP) Mailers" on page 68
- "UNIX-to-UNIX Copy Program (UUCP) Mailers" on page 69

For additional information about mailers, see `http://www.sendmail.org/m4/readme.html` or `/etc/mail/cf/README`.

Simple Mail Transfer Protocol (SMTP) Mailers

SMTP is the standard mail protocol that is used on the Internet. This protocol defines these mailers.

- `smtp` provides regular SMTP transfers to other servers.
- `esmtp` provides extended SMTP transfers to other servers.
- `smtp8` provides SMTP transfers to other servers without converting 8-bit data to MIME.
- `dsmtp` provides on-demand delivery by using the `F=%` mailer flag. Refer to "Changes to the `MAILER()` Declaration From Version 8.12 of `sendmail`" on page 115 and "Additional Delivery Agent Flags From Version 8.12 of `sendmail`" on page 116.

UNIX-to-UNIX Copy Program (UUCP) Mailers

If possible, avoid using UUCP. For an explanation, refer to `http://www.sendmail.org/m4/uucp_mailers.html` or do a search in `/etc/mail/cf/README` on this string: `USING UUCP MAILERS`.

UUCP defines these mailers.

`uucp-old` Names in the `$=U` class are sent to `uucp-old`. `uucp` is the obsolete name for this mailer. The `uucp-old` mailer uses an exclamation-point address in the headers.

`uucp-new` Names in the `$=Y` class are sent to `uucp-new`. Use this mailer when you know that the receiving UUCP mailer can manage multiple recipients in one transfer. `suucp` is the obsolete name for this mailer. The `uucp-new` mailer also uses an exclamation-point address in the headers.

If `MAILER(smtp)` is also specified in your configuration, two more mailers are defined.

`uucp-dom` This mailer uses domain-style addresses and, basically, applies the SMTP rewriting rules.

`uucp-uudom` Names in the `$=Z` class are sent to `uucp-uudom`. `uucp-uudom` and `uucp-dom` use the same header address format, domain-style addresses.

Note – Because the `smtp` mailer modifies the UUCP mailer, always put `MAILER(smtp)` before `MAILER(uucp)` in your `.mc` file.

Mail Addresses

The *mail address* contains the name of the recipient and the system to which the mail message is delivered. When you administer a small mail system that does not use a name service, addressing mail is easy. The login names uniquely identify the users. Complexity is introduced if you are administering a mail system that has more than one system with mailboxes or that has one or more domains. Additional complexity can be generated if you have a UUCP (or other) mail connection to servers outside your network. The information in the following sections can help you understand the parts and complexities of a mail address.

- "Domains and Subdomains" on page 69
- "Name Service Domain Name and Mail Domain Name" on page 70
- "Typical Format for Mail Addresses" on page 70
- "Route–Independent Mail Addresses" on page 71

Domains and Subdomains

Email addressing uses domains. A *domain* is a directory structure for network address naming. A domain can have one or more *subdomains*. The domain and subdomains of an address can be compared to the hierarchy of a file system. Just as a subdirectory is considered to be inside the directory above it, each subdomain in a mail address is considered to be inside the location to its right.

The following table shows some top-level domains.

TABLE 3–7 Top-Level Domains

Domain	Description
`com`	Commercial sites
`edu`	Educational sites
`gov`	United States government installations
`mil`	United States military installations
`net`	Networking organizations
`org`	Other nonprofit organizations

Domains are case insensitive. You can use uppercase, lowercase, or mixed-case letters in the domain part of an address without making any errors.

Name Service Domain Name and Mail Domain Name

When you are working with name service domain names and mail domain names, remember the following.

- By default, the `sendmail` program strips the first component from the NIS domain name to form the mail domain name. For example, if an NIS domain name were `bldg5.example.com`, its mail domain name would be `example.com`.
- Although mail domain addresses are case insensitive, the NIS domain name is not. For the best results, use lowercase characters when setting up the mail and NIS domain names.
- The DNS domain name and the mail domain name must be identical.

For more information, refer to "Interactions of `sendmail` With Name Services" on page 93.

Typical Format for Mail Addresses

Typically, a mail address has the following format. For further details, refer to "Route–Independent Mail Addresses" on page 71.

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

The part of the address to the left of the `@` sign is the local address. The local address can contain the following.

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

Note – The receiving mailer is responsible for determining what the local part of the address means. For information about mailers, refer to "Mailers and `sendmail`" on page 68.

The part of the address to the right of the `@` sign shows the domain levels, which is where the local address resides. A dot separates each subdomain. The domain part of the address can be an organization, a physical area, or a geographic region. Furthermore, the order of domain information is hierarchical, so the more local the subdomain, the closer the subdomain is to the `@` sign.

Route–Independent Mail Addresses

Mail addresses can be route independent. Route-independent addressing requires the sender of an email message to specify the name of the recipient and the final destination. A high-speed network, such as the Internet, uses route-independent addresses. Route-independent addresses can have this format.

user`@`*host*`.`*domain*

Route-independent addresses for UUCP connections can have this address format.

host`.`*domain*`!`*user*

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

user`@`*domain*

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

Mailbox Files

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

Mail should always be delivered to a local file system so that the user agent can pull mail from the mail spool and store it readily in the local mailbox. Do not use NFS-mounted file systems as the destination for a user's mailbox. Specifically, do not direct mail to a mail client that is mounting the `/var/mail` file system from a remote server. Mail for the user, in this instance, should be addressed to the mail server and not to the client host name. NFS-mounted file systems can cause problems with mail delivery and handling.

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

The following table shows some common naming conventions for special-purpose mailboxes.

TABLE 3–8 Conventions for the Format of Mailbox Names

Format	Description
username	User names are frequently the same as mailbox names.
Firstname`.`*Lastname* *Firstname_Lastname* *Firstinitial*`.`*Lastname* *Firstinitial_Lastname*	User names can be identified as full names with a dot (or an underscore) that separates the first and last names. Alternately, user names can be identified by a first initial with a dot (or an underscore) that separates the initial and the last name.
`postmaster`	Users can address questions and report problems with the mail system to the `postmaster` mailbox. Each site and domain should have a `postmaster` mailbox.
`MAILER-DAEMON`	`sendmail` automatically routes any mail that is addressed to the `MAILER-DAEMON` to the postmaster.
aliasname`-request`	Names that end in `-request` are administrative addresses for distribution lists. This address should redirect mail to the person who maintains the distribution list.
`owner-`*aliasname*	Names that begin with `owner-` are administrative addresses for distribution lists. This address should redirect mail to the person who handles mail errors.
`owner-owner`	This alias is used when no `owner-`*aliasname* alias exists for errors to be returned to. This address should redirect mail to the person who handles mail errors. This address also should be defined on any system that maintains a large number of aliases.
local`%`*domain*	The percent sign (`%`) marks a local address that is expanded when the message arrives at its destination. Most mail systems interpret mailbox names with `%` characters as full mail addresses. The `%` is replaced with an `@`, and the mail is redirected accordingly. Although many people use the `%` convention, this convention is not a formal standard. This convention is referred to as the “percent hack.” This feature is often used to help debug mail problems.

Starting with `sendmail` version 8, the envelope sender for mail that is sent to a group alias has been changed to the address that is expanded from the owner alias, if an owner alias exists. This change enables any mail errors to be sent to the alias owner, rather than being returned to the sender. With this change, users notice that mail that was sent to an alias looks as if the mail came from the alias owner, when delivered. The following alias format helps with some of the problems that are associated with this change.

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

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

Mail Aliases

An *alias* is an alternate name. For email, you can use aliases to assign a mailbox location or to define mailing lists. For a task map, refer to "Administering Mail Alias Files (Task Map)" on page 42 in Chapter 2, "Mail Services (Tasks)." Also, you can refer to "Mail Alias Files" on page 88 in this chapter.

For large sites, the mail alias typically defines the location of a mailbox. Providing a mail alias is like providing a room number as part of the address for an individual at a large corporation that occupies multiple rooms. If you do not provide the room number, the mail is delivered to a central address. Without a room number, extra effort is required to determine where within the building the mail is to be delivered. So, the possibility of an error increases. For example, if two people who are named Kevin Smith are in the same building, only one of them might get mail. To correct the problem, each Kevin Smith should have a room number added to his address.

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

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

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

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

```
fred: fred@planets
```

When you create mail lists that include users outside your domain, create the alias with the user name and the domain name. For example, if you have a user who is named `smallberries` on system `privet`, in domain `example.com`, create the alias as `smallberries@example.com`. The email address of the sender is now automatically translated to a fully qualified domain name when mail goes outside the user's domain.

The following list describes methods for creating and administering mail alias files.

- You can create mail aliases for global use in the NIS `aliases` map or in local `/etc/mail/aliases` files. You can also create and administer mailing lists that use the same alias files.
- Depending on the configuration of your mail services, you can administer aliases by using the NIS name service to maintain a global `aliases` database. Otherwise, you could update all the local `/etc/mail/aliases` files to keep the aliases synchronized.
- Users can also create and use aliases. Users can create aliases either in their local `~/.mailrc` file, which only the user can use, or in their local `/etc/mail/aliases` file, which anyone can use. Users cannot normally create or administer NIS alias files.

Hardware Components

You can provide the three required elements of mail configuration in the same system or have separate systems provide these elements.

- "Mail Host" on page 74
- "Mail Server" on page 75
- "Mail Client" on page 75

When users are to communicate with networks outside your domain, you must also add a fourth element, a mail gateway. For more information, refer to "Mail Gateway" on page 76. The following sections describe each hardware component.

Mail Host

A *mail host* is the machine that you designate as the main mail machine on your network. A mail host is the machine to which other systems at the site forward mail that cannot be delivered. You designate a system as a mail host in the `hosts` database by adding the word `mailhost` to the right of the IP address in the local `/etc/hosts` file. Alternately, you can add the word `mailhost` similarly to the hosts file in the name service. For detailed task information, refer to "How to Set Up a Mail Host" on page 29 in Chapter 2, "Mail Services (Tasks)."

A good candidate for a mail host is a system that is configured as a router from your network to the Internet global network. For more information, refer to Chapter 1, "Solaris PPP 4.0 (Overview)," in *Managing Serial Networks Using UUCP and PPP in Oracle Solaris 11.1*, Chapter 10, "UUCP (Overview)," in *Managing Serial Networks Using UUCP and PPP in Oracle*

Solaris 11.1, and "Configuring an IPv4 Router" in *Configuring and Administering Oracle Solaris 11.1 Networks*. If no systems on your local network have a modem, designate a system as the mail host.

Some sites use standalone machines that are not networked in a time-sharing configuration. Specifically, the standalone machine serves terminals that are attached to its serial ports. You can set up electronic mail for this configuration by designating the standalone system as the mail host of a single-system network. Overview of the Hardware Components provides a figure that shows a typical email configuration.

Mail Server

A *mailbox* is a single file that contains email for a particular user. Mail is delivered to the system where the user's mailbox resides, which can be on a local machine or a remote server. A *mail server* is any system that maintains user mailboxes in its `/var/mail` directory. For task information, refer to "How to Set Up a Mail Server" on page 26 in Chapter 2, "Mail Services (Tasks)."

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

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

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

By establishing a mail server for all mailboxes, you can simplify your process of doing backups. Backups can be difficult to do when mail is spread over many systems. The disadvantage of storing many mailboxes on one server is that the server can be a single point of failure for many users. However, the advantages of providing good backups usually make the risk worthwhile.

Mail Client

A mail client is a user of mail services with a mailbox on a mail server. Additionally, the mail client has a mail alias in the `/etc/mail/aliases` file that points to the location of the mailbox. For task information, refer to "How to Set Up a Mail Client" on page 27 in Chapter 2, "Mail Services (Tasks)."

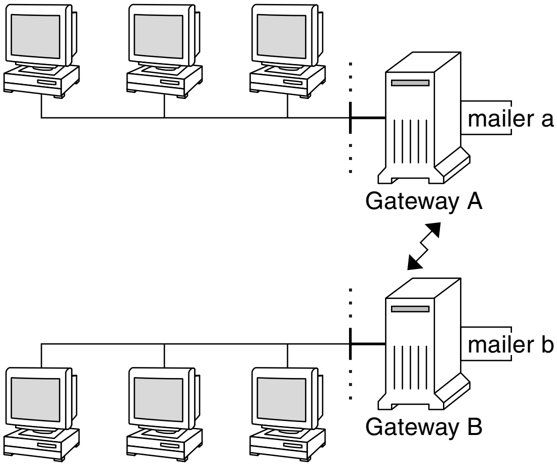
Mail Gateway

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

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

You can set up a mail gateway between two networks that use unmatched mailers, as shown in the next figure. To support this configuration, you must customize the `sendmail.cf` file on the mail gateway system, which can be a difficult and time-consuming process.

FIGURE 3–1 Gateway Between Different Communications Protocols



If you have a machine that provides connections to the Internet, you can configure that machine as the mail gateway. Carefully consider your site's security needs before you configure a mail gateway. You might need to create a firewall gateway between your corporate network and other networks, and set up that gateway as the mail gateway. For task information, refer to "How to Set Up a Mail Gateway" on page 31 in Chapter 2, "Mail Services (Tasks)."

Mail Service Programs and Files

Mail services include many programs and daemons that interact with each other. This section introduces the files, programs, terms, and concepts that are related to administering electronic mail.

Enhancement for vacation Utility

The vacation utility has been enhanced to enable a user to specify which incoming messages receive autogenerated replies. With this enhancement the user can avoid sharing confidential or contact information with unknown people. Messages from spammers or unknown people would not receive a reply.

This enhancement works by matching an incoming sender's email address to a list of domains or email addresses in a .vacation.filter file. This file is created by the user and is in the user's home directory. If a domain or email address match is found, a reply is sent. If no match is found, no reply is sent.

The .vacation.filter might contain entries such as these:

```
company.com
mydomain.com
onefriend@hisisp.com
anotherfriend@herisp.com
```

Note that each line contains one domain or one email address. Each entry must be on a separate line. For a sender's email address to match with an email address entry, the match must be exact, except for case. Whether the letters in the sender's address are lowercase or uppercase is ignored. For a sender's email address to match with a domain entry, the sender's address must contain the listed domain. For example, both somebody@dept.company.com and someone@company.com would be a match for a domain entry of company.com.

For more information, see the vacation(1) man page.

Contents of the /usr/bin Directory

The following table shows the contents of the /usr/bin directory, which is used for mail services.

Name	Type	Description
mail	File	A user agent.
mailcompat	File	A filter to store mail in SunOS 4.1 mailbox format.
mailq	File	A program that lists the content of the mail queue.
mailstats	File	A program that is used to read mail statistics that are stored in the /etc/mail/statistics file (if present).
mailx	File	A user agent.
mconnect	File	A program that connects to the mailer for address verification and debugging.
praliases	File	A command to "uncompile" the alias database. Refer to the uncompile information that is provided in the man page for praliases(1).
rmail	Symbolic Link	A symbolic link to /usr/bin/mail. Command that is often used to permit only the sending of mail.
vacation	File	A command to set up an automatic reply to mail.

Contents of the /etc/mail Directory

The following table shows the contents of the /etc/mail directory.

Name	Type	Description
Mail.rc	File	Default settings for the mailx user agent.
aliases	File	Mail-forwarding information.
aliases.db	File	Default binary form of mail-forwarding information that is created by running newaliases.
aliases.dir	File	Binary form of mail-forwarding information that is created by running newaliases. Can still be used, but is no longer used by default starting with the Solaris 9 release.
aliases.pag	File	Binary form of mail-forwarding information that is created by running newaliases. Can still be used, but is no longer used by default starting with the Solaris 9 release.

Name	Type	Description
mailx.rc	File	Default settings for the mailx user agent.
main.cf	Symbolic link	A symbolic link from this sample configuration file for main systems to sendmail.cf is provided for backwards compatibility. This file is not needed in version 8.13 of sendmail.
relay-domains	File	List of all domains for which relaying is allowed. By default, only the local domain is allowed.
sendmail.cf	File	Configuration file for mail routing.
submit.cf	File	New configuration file for the mail submission program (MSP). For more information, refer to "submit.cf Configuration File From Version 8.12 of sendmail" on page 106.
local-host-names	File	Optional file that you can create if the number of aliases for the mail host is too long.
helpfile	File	Help file that is used by the SMTP HELP command.
sendmail.pid	File	File that lists the PID of the listening daemon and is now in /system/volatile.
statistics	File	sendmail statistics file. If this file is present, sendmail logs the amount of traffic through each mailer. Previously, this file was called sendmail.st.
subsidiary.cf	Symbolic link	A symbolic link from this sample configuration file for subsidiary systems to sendmail.cf is provided for backwards compatibility. This file is not needed in version 8.13 of sendmail.
trusted-users	File	File that lists the users (one user per line) who can be trusted to perform certain mail operations. By default, only root is in this file. Certain mail operations, when performed by untrusted users, result in the following warning, X-Authentication-Warning: header being added to a message.

Contents of the /etc/mail/cf Directory

Within the /etc/mail directory is a subdirectory, cf, that contains all of the necessary files to build a sendmail.cf file. The content of cf is shown in Table 3–9.

To support a read-only /usr file system, the content of the /usr/lib/mail directory has been moved to the /etc/mail/cf directory. Note, however, these exceptions. The shell scripts /usr/lib/mail/sh/check-hostname and /usr/lib/mail/sh/check-permissions are now in the /usr/sbin directory. See "Other Files Used for Mail Services" on page 82. For backward compatibility, symbolic links point to each file's new location.

TABLE 3–9 Contents of the /etc/mail/cf Directory Used for Mail Services

Name	Type	Description
README	File	Describes the configuration files.
cf/main.cf	Symbolic Link	This file name is linked to cf/sendmail.cf. This file used to be the main configuration file.
cf/main.mc	Symbolic Link	This file name is linked to cf/sendmail.mc. This file was the file used to create the main configuration file.
cf/Makefile	File	Provides rules for building new configuration files.
cf/submit.cf	File	Is the configuration file for the mail submission program (MSP), which is used to submit messages.
cf/submit.mc	File	Is the file used to build the submit.cf file. The file defines m4 macros for the mail submission program (MSP).
cf/sendmail.cf	File	Is the main configuration file for sendmail.
cf/sendmail.mc	File	Contains the m4 macros that are used to generate the sendmail.cf file.
cf/subsidiary.cf	Symbolic Link	This file name is linked to cf/sendmail.cf. This file used to be the configuration file for hosts that NFS-mount /var/mail from another host.
cf/subsidiary.mc	Symbolic Link	This file name is linked to cf/sendmail.mc. This file used to contain the m4 macros that were used to generate the subsidiary.cf file.
domain	Directory	Provides site-dependent subdomain descriptions.
domain/generic.m4	File	Is the generic domain file from Berkeley Software Distribution.
domain/solaris-antispam.m4	File	Is the domain file with changes that make sendmail function like the previous versions of sendmail. However, relaying is disabled completely, sender addresses with no host name are rejected, and unresolvable domains are rejected.

TABLE 3–9 Contents of the /etc/mail/cf Directory Used for Mail Services *(Continued)*

Name	Type	Description
domain/solaris-generic.m4	File	Is the default domain file with changes that make sendmail function like the previous versions of sendmail.
feature	Directory	Contains definitions of specific features for particular hosts. See README for a full description of the features.
m4	Directory	Contains site-independent include files.
mailer	Directory	Contains definitions of mailers, which include local, smtp, and uucp.
main-v7sun.mc	File	Obsolete: this file name is renamed to cf/sendmail.mc.
ostype	Directory	Describes various operating system environments.
ostype/solaris2.m4	File	Defines default local mailer as mail.local.
ostype/solaris2.ml.m4	File	Defines default local mailer as mail.local.
ostype/solaris2.pre5.m4	File	Defines local mailer as mail.
ostype/solaris8.m4	File	Defines local mailer as mail.local (in LMTP mode), enables IPv6, specifies /system/volatile as the directory for the sendmail.pid file.
subsidiary-v7sun.mc	File	Obsolete: this file name is renamed to cf/sendmail.mc.

Contents of the /usr/lib Directory

The following table shows the contents of the /usr/lib directory, which is used for mail services.

TABLE 3–10 Contents of the /usr/lib Directory

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

TABLE 3–10 Contents of the /usr/lib Directory *(Continued)*

Name	Type	Description
smrsh	File	Shell program (sendmail restricted shell) that uses the "|program" syntax of sendmail to restrict programs that sendmail can run to those programs listed in the /var/adm/sm.bin directory. Refer to the smrsh(1M) man page for recommendations about what to include in /var/adm/sm.bin. To enable, include this m4 command, FEATURE('smrsh'), in your mc file.
mail	symbolic link	A symbolic link points to the/etc/mail/cf directory. For more information, refer to "Contents of the /etc/mail/cf Directory" on page 79.

Other Files Used for Mail Services

Several other files and directories are used for mail services, as shown in Table 3–11.

TABLE 3–11 Other Files Used for Mail Services

Name	Type	Description
/etc/default/sendmail	File	Lists the environment variables for the startup script for sendmail.
/etc/shells	File	Lists the valid login shells.
/etc/mail/cf/sh	Directory	Contains shell scripts that are used by the m4 build process and migration aids.
/system/volatile/sendmail.pid	File	File that lists the PID of the listening daemon.
/usr/sbin/check-permissions	File	Checks permissions of :include: aliases and .forward files and their parent directory path for correct permissions.
/usr/sbin/check-hostname	File	Verifies that sendmail is able to determine the fully qualified host name.
/usr/sbin/editmap	File	Queries and edits single records in database maps for sendmail.
/usr/sbin/in.comsat	File	Mail notification daemon.
/usr/sbin/makemap	File	Builds binary forms of keyed maps.
/usr/sbin/newaliases	Symbolic Link	A symbolic link to /usr/lib/sendmail. Used to create the binary form of the alias database. Previously in /usr/bin.
/usr/sbin/syslogd	File	Error message logger, used by sendmail.

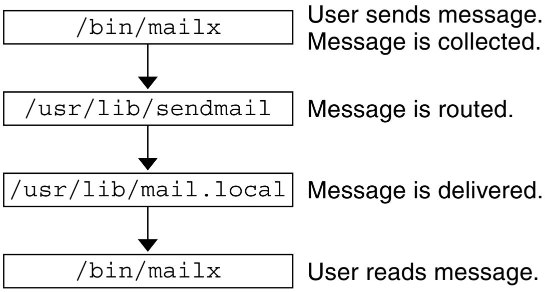
TABLE 3–11 Other Files Used for Mail Services (Continued)

Name	Type	Description
`/usr/sbin/etrn`	File	Perl script for starting the client-side remote mail queue.
`/var/mail/`*mailbox1*, `/var/mail/`*mailbox2*	File	Mailboxes for delivered mail.
`/var/spool/clientmqueue`	Directory	Storage for mail that is delivered by the client daemon.
`/var/spool/mqueue`	Directory	Storage for mail that is delivered by the master daemon.

Interactions of Mail Programs

Mail services are provided by a combination of the following programs, which interact as shown in the simplified illustration in Figure 3–2.

FIGURE 3–2 Interactions of Mail Programs



The following is a description of the interactions of mail programs.

1. Users send messages by using programs such as `mailx`. See the man page for `mailx(1)` for more information.
2. The message is collected by the program that generated the message, and the message is passed to the `sendmail` daemon.
3. The `sendmail` daemon *parses* the addresses (divides them into identifiable segments) in the message. The daemon uses information from the configuration file, `/etc/mail/sendmail.cf`, to determine network name syntax, aliases, forwarding information, and network topology. By using this information, `sendmail` determines which route a message must follow to get to a recipient.
4. The `sendmail` daemon passes the message to the appropriate system.

5. The /usr/lib/mail.local program on the local system delivers the mail to the mailbox in the /var/mail/*username* directory of the recipient of the message.
6. The recipient is notified that mail has arrived and retrieves the mail by using mail, mailx, or a similar program.

sendmail Program

The following list describes some of the capabilities of the sendmail program.

- sendmail can use different types of communications protocols, such as TCP/IP and UUCP.
- sendmail implements an SMTP server, message queuing, and mailing lists.
- sendmail controls name interpretation by using a pattern-matching system that can work with the following naming conventions.
  - Domain-based naming convention. The domain technique separates the issue of physical from logical naming. For more information about domains, refer to "Mail Addresses" on page 69.
  - Improvised techniques, such as providing network names that appear local to hosts on other networks.
  - Arbitrary (older) naming syntaxes.
  - Disparate naming schemes.

The Oracle Solaris operating system uses the sendmail program as a mail router. The following list describes some of its functions.

- sendmail is responsible for receiving and delivering email messages to a local delivery agent, such as mail.local or procmail.
- sendmail is a mail transfer agent that accepts messages from user agents, such as mailx and Mozilla Mail, and routes the messages through the Internet to their destination.
- sendmail controls email messages that users send in the following ways:
  - By evaluating the recipients' addresses
  - By choosing an appropriate delivery program
  - By rewriting the addresses in a format that the delivery agent can handle
  - By reformatting the mail headers as required
  - By finally passing the transformed message to the mail program for delivery

For more information about the sendmail program, refer to the following topics.

- "sendmail and Its Rerouting Mechanisms" on page 85
- "sendmail Features" on page 86
- "sendmail Configuration File" on page 87

sendmail and Its Rerouting Mechanisms

The sendmail program supports three mechanisms for mail rerouting. The mechanism that you choose depends on the type of change that is involved.

- A server change
- A domain-wide change
- A change for one user

Additionally, the rerouting mechanism that you choose can affect the level of administration that is required. Consider the following options.

1. One rerouting mechanism is *aliasing*.

   Aliasing can map names to addresses on a server-wide basis or a name service-wide basis, depending on the type of file that you use.

   Consider the following advantages and disadvantages of name service aliasing.

   - The use of a name service alias file permits mail rerouting changes to be administered from a single source. However, name service aliasing can create lag time when the rerouting change is propagated.
   - Name service administration is usually restricted to a select group of system administrators. A normal user would not administer this file.

   Consider the following advantages and disadvantages of using a server alias file.

   - By using a server alias file, rerouting can be managed by anyone who can become root on the designated server.
   - Server aliasing should create little or no lag time when the rerouting change is propagated.
   - The change only affects the local server, which might be acceptable if most of the mail is sent to one server. However, if you need to propagate this change to many mail servers, use a name service.
   - A normal user would not administer this change.

   For more information, refer to "Mail Alias Files" on page 88 in this chapter. For a task map, refer to "Administering Mail Alias Files (Task Map)" on page 42 in Chapter 2, "Mail Services (Tasks)."

2. The next mechanism is *forwarding*.

   This mechanism permits users to administer mail rerouting. Local users can reroute their incoming mail to the following.

   - Another mailbox
   - A different mailer
   - Another mail host

This mechanism is supported through the use of `.forward` files. For more information about these files, refer to “`.forward` Files” on page 90 in this chapter. For a task map, refer to “Administering `.forward` Files (Task Map)” on page 52 in Chapter 2, “Mail Services (Tasks).”

3. The last rerouting mechanism is *inclusion*.

   This mechanism allows users to maintain alias lists instead of requiring `root` access. To provide this feature, the `root` user must create an appropriate entry in the alias file on the server. After this entry is created, the user can reroute mail as necessary. For more information about inclusion, refer to “`/etc/mail/aliases` File” on page 88 in this chapter. For a task map, refer to “Administering Mail Alias Files (Task Map)” on page 42 in Chapter 2, “Mail Services (Tasks).”

Note – Programs that read mail, such as `/usr/bin/mailx`, can have aliases of their own, which are expanded before the message reaches `sendmail`. The aliases for `sendmail` can originate from a number of name service sources, such as local files or NIS. The order of the lookup is determined by the `svc:/system/name-service/switch` service. Refer to the `nsswitch.conf(4)` man page.

`sendmail` Features

The `sendmail` program provides the following features.

- `sendmail` is reliable. The program is designed to correctly deliver every message. No message should ever become completely lost.
- `sendmail` uses existing software for delivery whenever possible. For example, the user interacts with a mail-generating and a mail-sending program. When mail is submitted, the mail-generating program calls `sendmail`, which routes the message to the correct mailers. Because some of the senders might be network servers and some of the mailers might be network clients, `sendmail` can be used as an Internet mail gateway. See “Interactions of Mail Programs” on page 83 for a more detailed description of the process.
- `sendmail` can be configured to handle complex environments, including multiple networks. `sendmail` checks the contents of an address as well as its syntax to determine which mailer to use.
- `sendmail` uses configuration files to control mail configuration instead of requiring that configuration information be compiled into the code.
- Users can maintain their own mailing lists. Additionally, individuals can specify their own forwarding mechanism without modifying the domain-wide alias file, typically located in the domain-wide aliases that are maintained by NIS.
- Each user can specify a custom mailer to process incoming mail. The custom mailer can provide functions such as returning a message that reads: “I am on vacation.” See the `vacation(1)` man page for more information.
- `sendmail` batches addresses to a single host to reduce network traffic.

sendmail Configuration File

A *configuration file* controls the way that sendmail performs its functions. The configuration file determines the choice of delivery agents, address rewriting rules, and the format of the mail header. The sendmail program uses the information from the /etc/mail/sendmail.cf file to perform its functions.

The Oracle Solaris operating system provides two default configuration files in the /etc/mail directory.

1. sendmail.cf, a configuration file that is used to run sendmail in daemon mode.
2. submit.cf, a configuration file that is used to run sendmail in mail-submission program mode, instead of daemon mode. For more information, refer to "submit.cf Configuration File From Version 8.12 of sendmail" on page 106.

When setting up mail clients, mail servers, mail hosts, or mail gateways, consider the following:

- For mail clients or mail servers, you do not need to do anything to set up or edit the default configuration file.
- To set up a mail host or mail gateway, you need to set the relay mailer and relay host parameters that are needed for your mail configuration. For task information, refer to "Setting Up Mail Services (Task Map)" on page 25 or "Changing the sendmail Configuration" on page 33 in Chapter 2, "Mail Services (Tasks)." Note that with sendmail version 8.13, you no longer need the main.cf file.

The following list describes some configuration parameters that you can change, depending on the requirements of your site.

- Time values, which specify the following information.
  - Read timeouts.
  - Length of time a message remains undelivered in the queue before the message is returned to the sender. Refer to "Additional Queue Features From Version 8.12 of sendmail" on page 117. For a task map, refer to "Administering the Queue Directories (Task Map)" on page 48.
- Delivery modes, which specify how quickly mail is delivered.
- Load limits, which increase efficiency during busy periods. These parameters prevent sendmail from attempting to deliver large messages, messages to many recipients, and messages to sites that have been down for a long time.
- Log level, which specifies the kinds of problems that are logged.

Mail Alias Files

You can use any of the following files, maps, or tables to maintain aliases.

- "`.mailrc` Aliases" on page 88
- "`/etc/mail/aliases` File" on page 88
- "NIS `aliases` Map" on page 89

Your method of maintaining aliases depends on who uses the alias and who needs to be able to change the alias. Each type of alias has unique format requirements.

If you are looking for task information, refer to "Administering Mail Alias Files (Task Map)" on page 42 in Chapter 2, "Mail Services (Tasks)."

`.mailrc` Aliases

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

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

aliasname is the name that the user uses when sending mail, and *value* is a valid email address.

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

`/etc/mail/aliases` File

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

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

aliasname is the name that the user uses when sending mail to this alias, and *value* is a valid email address.

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

The aliases in the `/etc/mail/aliases` file are stored in text form. When you edit the `/etc/mail/aliases` file, you need to run the `newaliases` program. This program recompiles the database and makes the aliases available in binary form to the `sendmail` program. For task information, refer to "How to Set Up a Local Mail Alias File" on page 44 in Chapter 2, "Mail Services (Tasks)."

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

```
ignatz: ignatz@saturn
```

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

Normally, only the `root` user can edit this file. Another option is to create the following entry.

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

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

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

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

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

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

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

For a task map, refer to "Administering Mail Alias Files (Task Map)" on page 42 in Chapter 2, "Mail Services (Tasks)."

NIS aliases Map

All users in a local domain can use the entries that are in the NIS `aliases` map. The reason is that the `sendmail` program can use the NIS `aliases` map instead of the local `/etc/mail/aliases` files to determine mailing addresses. For more information, refer to the `nsswitch.conf(4)` man page.

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

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

aliasname is the name that the user uses when sending mail, and *value* is a valid email address.

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

aliasname: *aliasname@host*

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

For task information, refer to "How to Set Up an NIS `mail.aliases` Map" on page 43 in Chapter 2, "Mail Services (Tasks)."

`.forward` Files

Users can create a `.forward` file in their home directories that `sendmail`, along with other programs, can use to redirect mail or send mail. Refer to the following topics.

- "Situations to Avoid" on page 90
- "Controls for `.forward` files" on page 90
- "`.forward.`*hostname* File" on page 91
- "`.forward+`*detail* File" on page 91

For a task map, refer to "Administering `.forward` Files (Task Map)" on page 52 in Chapter 2, "Mail Services (Tasks)."

Situations to Avoid

The following list describes some situations that you can avoid or easily fix.

- If mail is not being delivered to the expected address, check the user's `.forward` file. The user might have put the `.forward` file in the home directory of `host1`, which forwards mail to `user@host2`. When the mail arrives at `host2`, `sendmail` checks for `user` in the NIS aliases and sends the message back to `user@host1`. This routing results in a loop and more bounced mail.
- To avoid security problems, never put `.forward` files in the `root` and `bin` accounts. If necessary, forward the mail by using the `aliases` file instead.

Controls for `.forward` files

For the `.forward` files to be an effective part of mail delivery, ensure that the following controls (mostly permissions settings) are correctly applied.

- The `.forward` file must be writable only by the owner of the file. This restriction prevents other users from breaking security.

- The paths that lead to the home directory must be owned and writable by `root` only. For example, if a `.forward` file is in `/export/home/terry`, `/export` and `/export/home` must be owned and writable by `root` only.
- The actual home directory should be writable only by the user.
- The `.forward` file cannot be a symbolic link, and this file cannot have more than one hard link.

`.forward.`*hostname* File

You can create a `.forward.`*hostname* file to redirect mail that is sent to a specific host. For example, if a user's alias has changed from `sandy@phoenix.example.com` to `sandy@example.com`, place a `.forward.phoenix` file in the home directory for `sandy`.

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

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

Sandy
```

In this example, mail can be forwarded to the correct place while the sender is notified of the alias change. Because the `vacation` program permits only one message file, you can forward only one message at a time. However, if the message is not host specific, one vacation message file can be used by `.forward` files for many hosts.

`.forward+`*detail* File

Another extension to the forwarding mechanism is the `.forward+`*detail* file. The *detail* string can be any sequence of characters except operator characters. The operator characters are `.:%&!^[]+`. By using this type of file, you can determine if someone else is using your email address without your knowledge. For instance, if a user tells someone to use the email address `sandy+test1@example.com`, the user would be able to identify any future mail that was delivered to this alias. By default, any mail that is sent to the `sandy+test1@example.com` alias is checked against the alias and the `.forward+`*detail* files. If no matches are made, the mail falls back to delivery to `sandy@example.com`, but the user is able to see a change in the `To:` mail header.

`/etc/default/sendmail` File

This file is used to store startup options for `sendmail` so that the options are not removed when a host is upgraded. The following variables can be used.

CLIENTOPTIONS="*string*"
: Selects additional options to be used with the client daemon, which looks in the client-only queue (`/var/spool/clientmqueue`) and acts as a client queue runner. No syntax checking is done, so be careful when making changes to this variable.

CLIENTQUEUEINTERVAL=#
: Similar to the `QUEUEINTERVAL` option, `CLIENTQUEUEINTERVAL` sets the time interval for mail queue runs. However, the `CLIENTQUEUEINTERVAL` option controls the functions of the client daemon, rather than the functions of the master daemon. Typically, the master daemon is able to deliver all messages to the SMTP port. However, if the message load is too high or the master daemon is not running, then messages go into the client-only queue, `/var/spool/clientmqueue`. The client daemon, which checks in the client-only queue, then acts as a client queue processor.

ETRN_HOSTS="*string*"
: Enables an SMTP client and server to interact immediately without waiting for the queue run intervals, which are periodic. The server can immediately deliver the portion of its queue that goes to the specified hosts. For more information, refer to the `etrn`(1M) man page.

MODE=-bd
: Selects the mode to start `sendmail` with. Use the `-bd` option or leave it undefined.

OPTIONS=*string*
: Selects additional options to be used with the master daemon. No syntax checking is done, so be careful when making changes to this variable.

QUEUEINTERVAL=#
: Sets the interval for mail queue runs on the master daemon. `#` can be a positive integer that is followed by either `s` for seconds, `m` for minutes, `h` for hours, `d` for days, or `w` for weeks. The syntax is checked before `sendmail` is started. If the interval is negative or if the entry does not end with an appropriate letter, the interval is ignored and `sendmail` starts with a queue interval of 15 minutes.

QUEUEOPTIONS=p
: Enables one persistent queue runner that sleeps between queue run intervals, instead of a new queue runner for each queue run interval. You can set this option to `p`, which is the only setting available. Otherwise, this option is not set.

Mail Addresses and Mail Routing

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

You can set up a client system to receive mail locally. Receiving mail locally is known as running `sendmail` in local mode. Local mode is the default for all mail servers and some clients. On a mail server or a mail client in local mode, a mail message is routed the following way.

Note – The following example assumes that you are using the default rule set in the sendmail.cf file.

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

   The mail address is expanded by checking for the mail alias in the name service and substituting the new value, if a new value is found. This new alias is then checked again.
2. If the mail is local, deliver the mail to /usr/lib/mail.local.

   The mail is delivered to a local mailbox.
3. If the mail address includes a host in this mail domain, deliver the mail to that host.
4. If the address does not include a host in this domain, forward the mail to the mail host.

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

Interactions of sendmail With Name Services

This section describes domain names as they apply to sendmail and name services. Furthermore, this section describes the rules for effective use of name services, and the specific interactions of sendmail with name services. For details, refer to the following topics.

- "sendmail.cf and Mail Domains" on page 93
- "sendmail and Name Services" on page 94
- "Interactions of NIS and sendmail" on page 95
- "Interactions of sendmail With NIS and DNS" on page 96

If you are looking for related task information, refer to "How to Use DNS With sendmail" on page 32 or "Administering Mail Alias Files (Task Map)" on page 42 in Chapter 2, "Mail Services (Tasks)."

sendmail.cf and Mail Domains

The standard sendmail.cf file uses mail domains to determine whether mail is delivered directly or through a mail host. Intradomain mail is delivered through a direct SMTP connection, while interdomain mail is forwarded to a mail host.

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

- The current host is not authorized to send packets directly to a host outside the mail domain.

- The mail host is capable of forwarding the mail to an authorized host that can transmit packets directly to an external host. Actually, the mail host can be an authorized host.

With these assumptions, the mail host is responsible for delivering or forwarding interdomain mail.

sendmail and Name Services

sendmail imposes various requirements on name services. To improve your understanding of these requirements, this section first describes the relationship of mail domains to name service domains. Then the section describes the various requirements. Refer to the following.

- "Mail Domains and Name Service Domains" on page 94
- "Requirements for Name Services" on page 94
- Man page for nsswitch.conf(4)

Mail Domains and Name Service Domains

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

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

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

Requirements for Name Services

This section describes the requirements that sendmail imposes on name services.

A host table or map in a name service must be set up to support three types of gethostbyname() queries.

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

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

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

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

For more information about the gethostbyname() function, refer to the gethostbyname(3NSL) man page.

Interactions of NIS and sendmail

The following list describes the interactions of sendmail and NIS and provides some guidance.

- **Mail domain name** – If you are setting up NIS as the primary name service, sendmail automatically strips the first component of the NIS domain name and uses the result as the mail domain name. For example, ebs.admin.acme.com becomes admin.acme.com.
- **Mail host name** – You must have a mailhost entry in the NIS host map.
- **Full host names** – The normal NIS setup does not "understand" the full host name. Rather than trying to make NIS understand the full host name, turn off this requirement from the sendmail side by editing the sendmail.cf file and replacing all occurrences of %l with %y. This change turns off sendmail's interdomain mail detection. If the target host can be resolved to an IP address, a direct SMTP delivery is attempted. Ensure that your NIS host map does not contain any host entry that is external to the current mail domain. Otherwise, you need to further customize the sendmail.cf file.
- **Matching full host names and short host names** – Follow the previous instructions about how to turn off gethostbyname() for a full host name.
- **Multiple NIS domains in one mail domain** – All NIS host maps under a common mail domain should have the same set of host entries. For example, the host map in the ebs.admin.acme.com domain should be the same as the host map in the esg.admin.acme.com. Otherwise, one address might work in one NIS domain, but fail in the other NIS domain.

For task information, refer to "Administering Mail Alias Files (Task Map)" on page 42 in Chapter 2, "Mail Services (Tasks)."

Interactions of `sendmail` With NIS and DNS

The following list describes the interactions of `sendmail` with NIS and DNS and provides some guidance.

- **Mail domain name** – If you are setting up NIS as the primary name service, `sendmail` automatically strips the first component of the NIS domain name and uses the result as the mail domain name. For example, `ebs.admin.acme.com` becomes `admin.acme.com`.
- **Mail host name** – When the DNS forwarding feature is turned on, queries that NIS cannot resolve are forwarded to DNS, so you do not need a `mailhost` entry in the NIS host map.
- **Full host names** – Although NIS does not "understand" full host names, DNS does understand. This requirement is satisfied when you follow the regular procedure for setting up NIS and DNS.
- **Matching full host names and short host names** – For every host entry in the NIS host table, you must have a corresponding host entry in DNS.
- **Multiple NIS domains in one mail domain** – All NIS host maps under a common mail domain should have the same set of host entries. For example, the host map in the `ebs.admin.acme.com` domain should be the same as the host map in the `esg.admin.acme.com` domain. Otherwise, one address might work in one NIS domain, but fail in the other NIS domain.

For task information, refer to "How to Use DNS With `sendmail`" on page 32 and "Administering Mail Alias Files (Task Map)" on page 42 in Chapter 2, "Mail Services (Tasks)."

Changes in Version 8.14 of `sendmail`

The sendmail service has been updated to version 8.14. In addition, here are some of the significant changes to sendmail.

- The system can be configured to automatically rebuild the `sendmail.cf` and the `submit.mc` configuration files. The required steps are documented in "How to Automatically Rebuild a Configuration File" on page 35.
- By default, the sendmail daemon runs in the new local daemon mode. The local–only mode only accepts incoming mail from the local host, for instance, mail from a cron job or between local users. Outbound mail is routed as expected, only the incoming mail is changed. The `-bl` option is used to select the local-only mode, also known as the Become Local mode. For more information about this mode, see the `sendmail`(1M) man page. For instructions on how to change back to the `-bd` or Become Daemon mode, see "How to Use `sendmail` in the Open Mode" on page 36.

- The -t and -u options to the makemap command now work as expected. The delimiter declared with the -t option is used as the delimiter, even with the -u option. Previously a space would be used as a delimiter if the -u option was used, regardless of the delimiter defined by the -t option. See the makemap(1M) man page for more information about these options.

Changes in Version 8.13 of sendmail

Although this version of sendmail provides many new features, the FallBackSmartHost option is the most significant addition. Because of this option you no longer need to use main.cf and subsidiary.cf. The main.cf file was used in environments that supported MX records. The subsidiary.cf file was used in environments without a fully operative DNS. In such environments a smart host was used instead of MX records. The FallBackSmartHost option provides unified configuration. It operates like an MX record of last possible preference for all environments. To ensure that mail gets delivered to clients, this option, if enabled, provides a well-connected (or smart) host that serves as a backup (or failover) for MX records that fail.

For more information about version 8.13, see the following sections:

- "Additional Command-Line Options in Version 8.13 of sendmail" on page 102
- "Additional and Revised Configuration File Options in Version 8.13 of sendmail" on page 103
- "Additional and Revised FEATURE() Declarations in Version 8.13 of sendmail" on page 104

Additionally, SMTP can run with Transport Layer Security (TLS). See the following description.

Support for Running SMTP With TLS in Version 8.13 of sendmail

Communications between SMTP servers and clients are not usually controlled or trusted on either end. This lack of security might allow a third party to monitor and even alter a communication between a server and a client. SMTP can use Transport Layer Security (TLS) in version 8.13 of sendmail to resolve this problem. This extended service to SMTP servers and clients provides the following:

- Private, authenticated communications over the Internet
- Protection from eavesdroppers and attackers

Note – The implementation of TLS is based on the Secure Sockets Layer (SSL) protocol.

STARTTLS is the SMTP keyword that initiates a secure SMTP connection by using TLS. This secure connection might be between two servers or between a server and a client. A secure connection is defined as follows:

- The source email address and the destination address are encrypted.
- The content of the email message is encrypted.

When the client issues the STARTTLS command, the server responds with one of the following:

- `220 Ready to start TLS`
- `501 Syntax error (no parameters allowed)`
- `454 TLS not available due to temporary reason`

The 220 response requires the client to start the TLS negotiation. The 501 response notes that the client incorrectly issued the STARTTLS command. STARTTLS is issued with no parameters. The 454 response necessitates that the client apply rule set values to determine whether to accept or maintain the connection.

Note that to maintain the Internet's SMTP infrastructure, publicly used servers must not require a TLS negotiation. However, a server that is used privately might require the client to perform a TLS negotiation. In such instances, the server returns this response:

```
530 Must issue a STARTTLS command first
```

The 530 response instructs the client to issue the STARTTLS command to establish a connection.

The server or client can refuse a connection if the level of authentication and privacy is not satisfactory. Alternately, because most SMTP connections are not secure, the server and client might maintain an unsecure connection. Whether to maintain or refuse a connection is determined by the configuration of the server and the client.

Support for running SMTP with TLS is not enabled by default. TLS is enabled when the SMTP client issues the STARTTLS command. Before the SMTP client can issue this command, you must set up the certificates that enable sendmail to use TLS. See “How to Set SMTP to Use TLS” on page 37. Note that this procedure includes defining new configuration file options and rebuilding your sendmail.cf file.

Configuration File Options for Running SMTP With TLS

The following table describes the configuration file options that are used to run SMTP with TLS. If you declare any of these options, use one of the following syntaxes:

- O *OptionName=argument* # for the configuration file
- -O *OptionName=argument* # for the command line
- define(‘*m4Name*',*argument*) # for m4 configuration

TABLE 3–12 Configuration File Options for Running SMTP With TLS

Option	Description
CACertFile	m4 name: confCACERT Argument: *filename* Default value: undefined Identifies the file that contains one CA certificate.
CACertPath	m4 name: confCACERT_PATH Argument: *path* Default value: undefined Identifies the path to the directory that contains certificates of CAs.
ClientCertFile	m4 name: confCLIENT_CERT Argument: *filename* Default value: undefined Identifies the file that contains the certificate of the client. Note that this certificate is used when sendmail acts as a client.
ClientKeyFile	m4 name: confCLIENT_KEY Argument: *filename* Default value: undefined Identifies the file that contains the private key that belongs to the client certificate.
CRLFile	m4 name: confCRL Argument: *filename* Default value: undefined Identifies the file that contains the certificate revocation status, which is used for X.509v3 authentication.
DHParameters	m4 name: confDH_PARAMETERS Argument: *filename* Default value: undefined Identifies the file that contains the Diffie-Hellman (DH) parameters.

TABLE 3–12 Configuration File Options for Running SMTP With TLS *(Continued)*

Option	Description
RandFile	m4 name: confRAND_FILE Argument: file:*filename* or egd:*UNIX socket* Default value: undefined Uses the file: prefix to identify the file that contains random data or uses the egd: prefix to identify the UNIX socket. Note that because the Oracle Solaris OS supports the random number generator device, this option does not need to be specified. See the random(7D) man page.
ServerCertFile	m4 name: confSERVER_CERT Argument: *filename* Default value: undefined Identifies the file that contains the server's certificate. This certificate is used when sendmail acts as a server.
Timeout.starttls	m4 name: confTO_STARTTLS Argument: *amount of time* Default value: 1h Sets the amount of time the SMTP client waits for a response to the STARTTLS command.
TLSSrvOptions	m4 name: confTLS_SRV_OPTIONS Argument: V Default value: undefined Determines whether the server asks for a certificate from the client. If this option is set to V, no client verification is performed.

For sendmail to support SMTP's use of TLS, the following options must be defined:

- CACertPath
- CACertFile
- ServerCertFile
- ClientKeyFile

Other options are not required.

Macros for Running SMTP With TLS

The following table describes the macros that are used by the STARTTLS command.

TABLE 3–13 Macros for Running SMTP With TLS

Macro	Description
`${cert_issuer}`	Holds the distinguished name (DN) of the certification authority (CA), which is the certificate issuer.
`${cert_subject}`	Holds the DN of the certificate that is called the **cert subject**.
`${cn_issuer}`	Holds the common name (CN) of the CA, which is the **cert issuer**.
`${cn_subject}`	Holds the CN of the certificate that is called the **cert subject**.
`${tls_version}`	Holds the version of TLS that is used for the connection.
`${cipher}`	Holds a set of cryptographic algorithms (known as a **cipher suite**) that is used for the connection.
`${cipher_bits}`	Holds in bits the key length of the symmetric encryption algorithm that is used for the connection.
`${verify}`	Holds the result of the verification of the certificate that was presented. Possible values are as follows: ■ `OK` – The verification succeeded. ■ `NO` – No certificate was presented. ■ `NOT` – No certificate was requested. ■ `FAIL` – The certificate that was presented could not be verified. ■ `NONE` – `STARTTLS` has not been performed. ■ `TEMP` – Temporary error occurred. ■ `PROTOCOL` – SMTP error occurred. ■ `SOFTWARE` – `STARTTLS` handshake failed.
`${server_name}`	Holds the name of the server with the current outgoing SMTP connection.
`${server_addr}`	Holds the address of the server with the current outgoing SMTP connection.

Rule Sets for Running SMTP With TLS

The following table describes rule sets that determine whether an SMTP connection that uses TLS should be accepted, continued, or refused.

TABLE 3–14 Rule Sets for Running SMTP With TLS

Rule Set	Description
`tls_server`	Acting as a client, `sendmail` uses this rule set to determine whether the server is currently supported by TLS.
`tls_client`	Acting as a server, `sendmail` uses this rule set to determine whether the client is currently supported by TLS.

TABLE 3–14 Rule Sets for Running SMTP With TLS (Continued)

Rule Set	Description
tls_rcpt	This rule set requires verification of the recipient's MTA. This recipient restriction makes attacks such as DNS spoofing impossible.
TLS_connection	This rule set checks the requirement that is specified by the RHS of the access map against the actual parameters of the current TLS connection.
try_tls	sendmail uses this rule set to determine the feasibility of using STARTTLS when connecting to another MTA. If the MTA cannot properly implement STARTTLS, then STARTTLS is not used.

For more information, see http://www.sendmail.org/m4/starttls.html.

Security Considerations Related to Running SMTP With TLS

As a standard mail protocol that defines mailers that run over the Internet, SMTP is not an end-to-end mechanism. Because of this protocol limitation, TLS security through SMTP does not include mail user agents. Mail user agents act as an interface between users and a mail transfer agent such as sendmail.

Also, mail might be routed through multiple servers. For complete SMTP security the entire chain of SMTP connections must have TLS support.

Finally, the level of negotiated authentication and privacy between each pair of servers or a client and server pair must be considered. For more information, see "Authentication Services" in *Oracle Solaris 11.1 Administration: Security Services*.

Additional Command-Line Options in Version 8.13 of sendmail

The following table describes additional command-line options that are available in version 8.13 of sendmail. Other command-line options are described in the sendmail(1M) man page.

TABLE 3–15 Command-Line Options Available in Version 8.13 of sendmail

Option	Description
-D *logfile*	Sends debugging output to the indicated *logfile*, instead of including this information with the standard output.
-q[!]Q*substr*	Specifies the processing of quarantined jobs that have this *substr*, which is a substring of the quarantine *reason*. See the description of the -Q*reason* option. If ! is added, this option processes quarantined jobs that do not have this *substr*.

TABLE 3–15 Command-Line Options Available in Version 8.13 of `sendmail` *(Continued)*

Option	Description
`-Q`*reason*	Quarantines a normal queue item with this *reason*. If no *reason* is given, the quarantined queue item is unquarantined. This option works with the `-q[!]Q`*substr* option. The *substr* is a portion (or substring) of the *reason*.

Additional and Revised Configuration File Options in Version 8.13 of `sendmail`

The following table describes the added and revised configuration file options. If you declare any of these options, use one of the following syntaxes.

```
O OptionName=argument           # for the configuration file
-O OptionName=argument          # for the command line
define(‘m4Name',argument)      # for m4 configuration
```

TABLE 3–16 Configuration File Options Available in Version 8.13 of `sendmail`

Option	Description
`ConnectionRateWindowSize`	`m4` name: `confCONNECTION_RATE_WINDOW_SIZE` Argument: *number* Default value: `60` Sets the number of seconds for incoming connections to be maintained.
`FallBackSmartHost`	`m4` name: `confFALLBACK_SMARTHOST` Argument: *hostname* To ensure that mail gets delivered to the clients, this option provides a well-connected host that serves as a backup (or failover) for MX records that fail.
`InputMailFilters`	`m4` name: `confINPUT_MAIL_FILTERS` Argument: *filename* Lists the input mail filters for the `sendmail` daemon.
`PidFile`	`m4` name: `confPID_FILE` Argument: *filename* Default value: `/system/volatile/sendmail.pid` As in previous releases, the file name is macro-expanded before it is opened. Additionally, in version 8.13, the file is unlinked when `sendmail` exits.

TABLE 3–16 Configuration File Options Available in Version 8.13 of sendmail *(Continued)*

Option	Description
QueueSortOrder	m4 name: confQUEUE_SORT_ORDER Added argument: none In version 8.13 none is used to specify no sorting order.
RejectLogInterval	m4 name: confREJECT_LOG_INTERVAL Argument: *period-of-time* Default value: 3h, which represents three hours. When a daemon connection is refused for the *period-of-time* specified, the information is logged.
SuperSafe	m4 name: confSAFE_QUEUE Short name: s Added argument: postmilter Default value: true If postmilter is set, sendmail defers synchronizing the queue file until all milters have signaled acceptance of the message. For this argument to be useful, sendmail must be running as an SMTP server. Otherwise, postmilter operates as if you are using the true argument.

Additional and Revised FEATURE() Declarations in Version 8.13 of sendmail

The following table describes the added and revised FEATURE() declarations. This m4 macro uses the following syntax.

FEATURE(‘*name*’, ‘*argument*’)

TABLE 3–17 FEATURE() Declarations Available in Version 8.13 of sendmail

Name of FEATURE()	Description
conncontrol	Works with the access_db rule set to check the number of incoming SMTP connections. For details, see /etc/mail/cf/README.
greet_pause	Adds the greet_pause rule set, which enables open proxy and SMTP slamming protection. For details, see /etc/mail/cf/README.

TABLE 3–17 FEATURE() Declarations Available in Version 8.13 of sendmail *(Continued)*

Name of FEATURE()	Description
local_lmtp	The default argument continues to be mail.local, which is the LMTP-capable mailer in this Oracle Solaris release. However, in version 8.13, if a different LMTP-capable mailer is used, its path name can be specified as a second parameter and the arguments that are passed to the second parameter can be specified in the third parameter. For example: FEATURE('local_lmtp', '/usr/local/bin/lmtp', 'lmtp')
mtamark	Provides experimental support for "Marking Mail Transfer Agents in Reverse DNS with TXT RRs" (MTAMark). For details, see /etc/mail/cf/README.
ratecontrol	Works with the access_db rule set to control connection rates for hosts. For details, see /etc/mail/cf/README.
use_client_ptr	If this FEATURE() is enabled, the rule set check_relay overrides its first argument with this argument, $&{client_ptr}.

Changes From Version 8.12 of sendmail

This section contains information about the following topics.

Support for TCP Wrappers From Version 8.12 of sendmail

TCP wrappers provide a way of implementing access controls by checking the address of a host requesting a particular network service against an access control list (ACL). Requests are granted or denied, accordingly. Besides providing this access control mechanism, TCP wrappers also log host requests for network services, which is a useful monitoring function. Examples of network services that might be placed under access control include rlogind, telnetd, and ftpd.

Starting with version 8.12, sendmail enables the use of TCP wrappers. This check does not bypass other security measures. By enabling TCP wrappers in sendmail, a check has been added to validate the source of a network request before the request is granted. See the hosts_access(4) man page.

Note – Support for TCP wrappers in inetd(1M) and sshd(1M) started with the Solaris 9 release.

For information about ACLs, see "Using Access Control Lists to Protect UFS Files" in *Oracle Solaris 11.1 Administration: Security Services*.

submit.cf Configuration File From Version 8.12 of sendmail

Starting with version 8.12, sendmail includes an additional configuration file, /etc/mail/submit.cf. This file, submit.cf, is used to run sendmail in mail-submission program mode instead of daemon mode. Mail-submission program mode, unlike daemon mode, does not require root privilege, so this new paradigm provides better security.

See the following list of functions for submit.cf:

- sendmail uses submit.cf to run in mail-submission program (MSP) mode, which submits email messages and can be started by programs (such as mailx), as well as by users. Refer to the descriptions of the -Ac option and the -Am option in the sendmail(1M) man page.
- submit.cf is used in the following operating modes:
  - -bm, which is the default operating mode
  - -bs, which uses standard input to run SMTP
  - -bt, which is the test mode that is used to resolve addresses
- sendmail, when using submit.cf, does not run as an SMTP daemon.

- sendmail, when using submit.cf, uses /var/spool/clientmqueue, the client-only mail queue, which holds messages that were not delivered to the sendmail daemon. Messages in the client-only queue are delivered by the client "daemon," which is really acting as a client queue runner.
- By default, sendmail uses submit.cf periodically to run the MSP queue (otherwise known as the client-only queue), /var/spool/clientmqueue.

  ```
  /usr/lib/sendmail -Ac -q15m
  ```

Note the following:

- Starting with the Solaris 9 release, submit.cf is provided automatically.
- submit.cf does not require any planning or preliminary procedures prior to the installation of the Solaris 9 release or a more recent release.
- Unless you specify a configuration file, sendmail automatically uses submit.cf as required. Basically, sendmail knows which tasks are appropriate for submit.cf and which tasks are appropriate for sendmail.cf.

Functions That Distinguish sendmail.cf From submit.cf

The sendmail.cf configuration file is for the daemon mode. When using this file, sendmail is acting as a mail transfer agent (MTA), which is started by root.

```
/usr/lib/sendmail -L sm-mta -bd -q1h
```

See the following list of other distinguishing functions for sendmail.cf:

- By default, sendmail.cf accepts SMTP connections on ports 25 and 587.
- By default, sendmail.cf runs the main queue, /var/spool/mqueue.

Functional Changes From Version 8.12 of sendmail

With the addition of submit.cf, the following functional changes have occurred:

- Starting with version 8.12 of sendmail, only root can run the mail queue. For further details, refer to the changes that are described in the mailq(1) man page. For new task information, refer to "Administering the Queue Directories (Task Map)" on page 48.
- The mail-submission program mode runs without root privilege, which might prevent sendmail from having access to certain files (such as the .forward files). Therefore, the -bv option for sendmail could give the user misleading output. No workaround is available.
- Prior to sendmail version 8.12, if you were not running sendmail in daemon mode, you would only prevent the delivery of inbound mail. Starting with sendmail version 8.12, if you are not running the sendmail daemon with the default configuration, you also prevent the delivery of outbound mail. The client queue runner (also known as the mail submission program) must be able to submit mail to the daemon on the local SMTP port. If the client queue runner tries to open an SMTP session with the local host and the daemon is not

listening on the SMTP port, the mail remains in the queue. The default configuration does run a daemon, so this problem does not occur if you are using the default configuration. However, if you have disabled your daemon, refer to "How to Manage Mail Delivery by Using an Alternate Configuration of sendmail.cf" on page 41 for a way to resolve this problem.

Additional or Deprecated Command-Line Options From Version 8.12 of sendmail

The following table describes additional or deprecated command-line options for sendmail. Other command-line options are described in the sendmail(1M) man page.

TABLE 3–18 Additional or Deprecated Command-Line Options From Version 8.12 of sendmail

Option	Description
-Ac	Indicates that you want to use the configuration file, submit.cf, even if the operation mode does not indicate an initial mail submission. For more information about submit.cf, refer to "submit.cf Configuration File From Version 8.12 of sendmail" on page 106.
-Am	Indicates that you want to use the configuration file, sendmail.cf, even if the operation mode indicates an initial mail submission. For more information, refer to "submit.cf Configuration File From Version 8.12 of sendmail" on page 106.
-bP	Indicates that you are printing the number of entries in each queue.
-G	Indicates that the message that is being submitted from the command line is for relaying, not for initial submission. The message is rejected if the addresses are not fully qualified. No canonicalization is done. As is noted in the Release Notes that are part of the sendmail distribution on ftp://ftp.sendmail.org, improperly formed messages might be rejected in future releases.
-L *tag*	Sets the identifier that is used for syslog messages to the supplied *tag*.
-q[!]I *substring*	Processes only jobs that contain this *substring* of one of the recipients. When ! is added, the option processes only jobs that do not have this *substring* of one of the recipients.
-q[!]R *substring*	Processes only jobs that contain this *substring* of the queue ID. When ! is added, the option processes only jobs that do not have this *substring* of the queue ID.
-q[!]S *substring*	Processes only jobs that contain this *substring* of the sender. When ! is added, the option processes only jobs that do not have this *substring* of the sender.
-qf	Processes saved messages in the queue once, without using the fork system call, and runs the process in the foreground. Refer to the fork(2) man page.
-qG*name*	Processes only the messages in the *name* queue group.
-qp*time*	Processes saved messages in the queue at a specific interval of time with a single child that is forked for each queue. The child sleeps between queue runs. This new option is similar to the -q*time*, which periodically forks a child to process the queue.

TABLE 3–18 Additional or Deprecated Command-Line Options From Version 8.12 of sendmail *(Continued)*

Option	Description
-U	As is noted in the Release Notes that are part of the sendmail distribution on ftp://ftp.sendmail.org, this option is not available as of version 8.12. Mail user agents should use the -G argument.

Additional Arguments for the PidFile and ProcessTitlePrefix Options From Version 8.12 of sendmail

The following table describes additional macro-processed arguments for the PidFile and ProcessTitlePrefix options. For more information about these options, see the sendmail(1M) man page.

TABLE 3–19 Arguments for the PidFile and ProcessTitlePrefix Options

Macro	Description
${daemon_addr}	Provides daemon address (for example, 0.0.0.0)
${daemon_family}	Provides daemon family (for example, inet, and inet6)
${daemon_info}	Provides daemon information (for example, SMTP+queueing@00:30:00)
${daemon_name}	Provides daemon name (for example, MSA)
${daemon_port}	Provides daemon port (for example, 25)
${queue_interval}	Provides queue run interval (for example, 00:30:00)

Additional Defined Macros From Version 8.12 of sendmail

The following table describes additional macros that are reserved for use by the sendmail program. The macros' values are assigned internally. For more information, refer to the sendmail(1M) man page.

TABLE 3–20 Additional Defined Macros for sendmail

Macro	Description
${addr_type}	Identifies the current address as an envelope sender or a recipient address.
${client_resolve}	Holds the result of the resolve call for ${client_name}: OK, FAIL, FORGED, or TEMP.

TABLE 3–20 Additional Defined Macros for sendmail *(Continued)*

Macro	Description
${deliveryMode}	Specifies the current delivery mode sendmail is using instead of the value of the DeliveryMode option.
${dsn_notify}, ${dsn_envid}, ${dsn_ret}	Holds the corresponding DSN parameter values.
${if_addr}	Provides the interface's address for the incoming connection if the interface does not belong to the loopback net. This macro is especially useful for virtual hosting.
${if_addr_out}, ${if_name_out}, ${if_family_out}	Avoids the reuse of ${if_addr}. Holds the following values respectively: The address of the interface for the outgoing connection The host name of the interface for the outgoing connection The family of the interface for the outgoing connection
${if_name}	Provides the interface's host name for the incoming connection and is especially useful for virtual hosting.
${load_avg}	Checks and reports the current average number of jobs in the run queue.
${msg_size}	Holds the value of the message size (SIZE=parameter) in an ESMTP dialogue before the message has been collected. Thereafter, the macro holds the message size as computed by sendmail and is used in check_compat. For information about check_compat, refer to Table 3–24.
${nrcpts}	Holds the number of validated recipients.
${ntries}	Holds the number of delivery attempts.
${rcpt_mailer}, ${rcpt_host}, ${rcpt_addr}, ${mail_mailer}, ${mail_host}, ${mail_addr}	Holds the results of parsing the RCPT and MAIL arguments, which is the resolved right-hand side (RHS) triplet from the mail delivery agent ($#*mailer*), the host ($@*host*), and the user ($:*addr*).

Additional Macros From Version 8.12 of sendmail

In this section, you can find a table that describes the additional macros that are used to build the sendmail configuration file.

TABLE 3–21 Additional Macros Used to Build the sendmail Configuration File

Macro	Description
LOCAL_MAILER_EOL	Overrides the default end-of-line string for the local mailer.
LOCAL_MAILER_FLAGS	Adds Return-Path: header by default.
MAIL_SETTINGS_DIR	Contains the path (including the trailing slash) for the mail settings directory.
MODIFY_MAILER_FLAGS	Improves the *_MAILER_FLAGS. This macro sets, adds, or deletes flags.
RELAY_MAILER_FLAGS	Defines additional flags for the relay mailer.

Additional MAX Macros From Version 8.12 of sendmail

Use the following macros to configure the maximum number of commands that can be received before sendmail slows its delivery. You can set these MAX macros at compile time. The maximum values in the following table also represent the current default values.

TABLE 3–22 Additional MAX Macros

Macro	Maximum Value	Commands Checked by Each Macro
MAXBADCOMMANDS	25	Unknown commands
MAXNOOPCOMMANDS	20	NOOP, VERB, ONEX, XUSR
MAXHELOCOMMANDS	3	HELO, EHLO
MAXVRFYCOMMANDS	6	VRFY, EXPN
MAXETRNCOMMANDS	8	ETRN

Note – You can disable a macro's check by setting the macro's value to zero.

Additional and Revised m4 Configuration Macros From Version 8.12 of sendmail

This section contains a table of additional and revised m4 configuration macros for sendmail. Use the following syntax to declare these macros.

symbolic-name(‘*value*’)

If you need to build a new `sendmail.cf` file, refer to "Changing the `sendmail` Configuration" on page 33 in Chapter 2, "Mail Services (Tasks)."

TABLE 3–23 Additional and Revised m4 Configuration Macros for `sendmail`

m4 Macro	Description
`FEATURE()`	For details, refer to "Changes to the `FEATURE()` Declaration From Version 8.12 of `sendmail`" on page 112.
`LOCAL_DOMAIN()`	This macro adds entries to class `w` (`$=w`).
`MASQUERADE_EXCEPTION()`	A new macro that defines hosts or subdomains that cannot be masqueraded.
`SMART_HOST()`	This macro can now be used for bracketed addresses, such as `user@[`*host*`]`.
`VIRTUSER_DOMAIN()` or `VIRTUSER_DOMAIN_FILE()`	When these macros are used, include `$={VirtHost}` in `$=R`. As a reminder, `$=R` is the set of host names that are allowed to relay.

Changes to the `FEATURE()` Declaration From Version 8.12 of `sendmail`

Refer to the following tables for information about the specific changes to the `FEATURE()` declarations.

To use the new and revised `FEATURE` names, use the following syntax.

```
FEATURE('name', 'argument')
```

If you need to build a new `sendmail.cf` file, refer to "Changing the `sendmail` Configuration" on page 33 in Chapter 2, "Mail Services (Tasks)."

TABLE 3–24 Additional and Revised `FEATURE()` Declarations

Name of `FEATURE()`	Description
`compat_check`	Argument: Refer to the example in the following paragraph. This new `FEATURE()` enables you to look for a key in the access map that consists of the sender address and the recipient address. This `FEATURE()` is delimited by the following string, <@>. *sender@sdomain*<@>*recipient@rdomain* is an example.

TABLE 3–24 Additional and Revised FEATURE() Declarations *(Continued)*

Name of FEATURE()	Description
delay_checks	Argument: friend, which enables a spam-friend test, or hater, which enables a spam-hater test. A new FEATURE() that delays all checks. By using FEATURE('delay_checks'), the rule sets check_mail and check_relay are not called when a client connects or issues a MAIL command respectively. Instead, these rule sets are called by the check_rcpt rule set. For details, refer to the /etc/mail/cf/README file.
dnsbl	Argument: This FEATURE() accepts a maximum of two arguments: ■ DNS server name ■ Rejection message A new FEATURE() that you can include multiple times to check the return values for DNS lookups. Note that this FEATURE() enables you to specify the behavior of temporary lookup failures.
enhdnsbl	Argument: domain name. A new FEATURE() that is an enhanced version of dnsbl, which enables you to check the return values for DNS lookups. For more information, refer to /etc/mail/cf/README.
generics_entire_domain	Argument: None. A new FEATURE() that you can also use to apply genericstable to subdomains of $=G.
ldap_routing	Argument: For details, refer to the “Release Notes” in http://www.sendmail.org. A new FEATURE() that implements LDAP address routing.
local_lmtp	Argument: Path name of an LMTP-capable mailer. The default is mail.local, which is LMTP capable in this Oracle Solaris release. A FEATURE() that now sets the delivery status notification (DSN) diagnostic-code type for the local mailer to the proper value of SMTP.
local_no_masquerade	Argument: None. A new FEATURE() that you can use to avoid masquerading for the local mailer.
lookupdotdomain	Argument: None. A new FEATURE() that you can also use to look up the *.domain* in the access map.

TABLE 3–24 Additional and Revised FEATURE() Declarations *(Continued)*

Name of FEATURE()	Description
nocanonify	Argument: canonify_hosts or nothing. A FEATURE() that now includes the following features. Enables a list of domains, as specified by CANONIFY_DOMAIN or CANONIFY_DOMAIN_FILE, to be passed to the $[and $] operators for canonification. Enables addresses that have only a host name, such as <user@host>, to be canonified, if canonify_hosts is specified as its parameter. Adds a trailing dot to addresses with more than one component.
no_default_msa	Argument: None. A new FEATURE() that turns off sendmail's default setting from m4–generated configuration files to "listen" on several different ports, an implementation of RFC 2476.
nouucp	Argument: reject, which does not allow the ! token, or nospecial, which does allow the ! token. A FEATURE() that determines whether to allow the ! token in the local part of an address.
nullclient	Argument: None. A FEATURE() that now provides the full rule sets of a normal configuration, allowing antispam checks to be performed.
preserve_local_plus_detail	Argument: None. A new FEATURE() that enables you to preserve the +detail portion of the address when sendmail passes the address to the local delivery agent.
preserve_luser_host	Argument: None. A new FEATURE() that enables you to preserve the name of the recipient host, if LUSER_RELAY is used.
queuegroup	Argument: None. A new FEATURE() that enables you to select a queue group that is based on the full email address or on the domain of the recipient.
relay_mail_from	Argument: The *domain* is an optional argument. A new FEATURE() that allows relaying if the mail sender is listed as a RELAY in the access map and is tagged with the From: header line. If the optional *domain* argument is given, the domain portion of the mail sender is also checked.

TABLE 3–24 Additional and Revised FEATURE() Declarations *(Continued)*

Name of FEATURE()	Description
virtuser_entire_domain	Argument: None. A FEATURE() that you can now use to apply $={VirtHost}, a new class for matching virtusertable entries that can be populated by VIRTUSER_DOMAIN or VIRTUSER_DOMAIN_FILE. FEATURE('virtuser_entire_domain') can also apply the class $={VirtHost} to entire subdomains.

The following FEATURE() declarations are no longer supported.

TABLE 3–25 Unsupported FEATURE() Declarations

Name of FEATURE()	Replacement
rbl	FEATURE('dnsbl') and FEATURE('enhdnsbl') replace this FEATURE(), which has been removed.
remote_mode	MASQUERADE_AS('$S') replaces FEATURE('remote_mode') in /etc/mail/cf/subsidiary.mc. $S is the SMART_HOST value in sendmail.cf.
sun_reverse_alias_files	FEATURE('genericstable').
sun_reverse_alias_nis	FEATURE('genericstable').
sun_reverse_alias_nisplus	FEATURE('genericstable').

Changes to the MAILER() Declaration From Version 8.12 of sendmail

The MAILER() declaration specifies support for delivery agents. To declare a delivery agent, use the following syntax.

MAILER('*symbolic-name*')

Note the following changes.

- In this new version of sendmail, the MAILER('smtp') declaration now includes an additional mailer, dsmtp, which provides on-demand delivery by using the F=% mailer flag. The dsmtp mailer definition uses the new DSMTP_MAILER_ARGS, which defaults to IPC $h.
- Numbers for rule sets that are used by MAILERs have been removed. You now have no required order for listing your MAILERs except for MAILER('uucp'), which must follow MAILER('smtp') if uucp-dom and uucp-uudom are used.

For more information about mailers, refer to "Mailers and sendmail" on page 68. If you need to build a new sendmail.cf file, refer to "Changing the sendmail Configuration" on page 33 in Chapter 2, "Mail Services (Tasks)."

Additional Delivery Agent Flags From Version 8.12 of sendmail

The following table describes additional delivery agent flags, which by default are not set. These single-character flags are Boolean. You can set or unset a flag by including or excluding it in the F= statement of your configuration file, as shown in the following example.

```
Mlocal,    P=/usr/lib/mail.local, F=lsDFMAw5:/|@qSXfmnz9, S=10/30, R=20/40,
Mprog,     P=/bin/sh, F=lsDFMoqeu9, S=10/30, R=20/40, D=$z:/,
Msmtp,     P=[IPC], F=mDFMuX, S=11/31, R=21, E=\r\n, L=990,
Mesmtp,    P=[IPC], F=mDFMuXa, S=11/31, R=21, E=\r\n, L=990,
Msmtp8,    P=[IPC], F=mDFMuX8, S=11/31, R=21, E=\r\n, L=990,
Mrelay,    P=[IPC], F=mDFMuXa8, S=11/31, R=61, E=\r\n, L=2040,
```

TABLE 3–26 Additional Mailer Flags

Flag	Description
%	Mailers that use this flag do not attempt delivery to the initial recipient of a message or to queue runs unless the queued message is selected by using an ETRN request or one of the following queue options: -qI, -qR, or -qS.
1	This flag disables the ability of the mailer to send null characters (for example, \0).
2	This flag disables the use of ESMTP and requires that SMTP be used instead.
6	This flag enables mailers to strip headers to 7 bit.

Additional Equates for Delivery Agents From Version 8.12 of sendmail

The following table describes additional equates that you can use with the M delivery-agent definition command. The following syntax shows you how to append new equates or new arguments to the equates that already exist in the configuration file.

```
Magent-name, equate, equate, ...
```

The following example includes the new W= equate. This equate specifies the maximum time to wait for the mailer to return after all data has been sent.

```
Msmtp, P=[IPC], F=mDFMuX, S=11/31, R=21, E=\r\n, L=990, W=2m
```

When you modify the definition of a value for m4 configuration, use the syntax that is provided in the following example.

```
define(‘SMTP_MAILER_MAXMSGS’, ‘1000’)
```

The preceding example places a limit of 1000 on the number of messages that are delivered per connection on an smtp mailer.

If you need to build a new sendmail.cf file, refer to "Changing the sendmail Configuration" on page 33 in Chapter 2, "Mail Services (Tasks)."

Note – Typically, you modify the equate definitions in the mailer directory only when you fine-tune.

TABLE 3–27 Additional Equates for Delivery Agents

Equate	Description
/=	Argument: Path to a directory Specifies a directory to apply chroot() to before the mailer program is executed
m=	Argument: Any of the following m4 values that have previously been defined with the define() routine SMTP_MAILER_MAXMSGS, for the smtp mailer LOCAL_MAILER_MAXMSGS, for the local mailer RELAY_MAILER_MAXMSGS, for the relay mailer Limits the number of messages that are delivered per connection on an smtp, local, or relay mailer
W=	Argument: An increment of time Specifies the maximum time to wait for the return of the mailer after all data has been sent

Additional Queue Features From Version 8.12 of sendmail

The following list provides details about additional queue features.

- This release supports multiple queue directories. To use multiple queues, supply a QueueDirectory option value in the configuration file that ends with an asterisk (*), as is shown in the following example.

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

  The option value, /var/spool/mqueue/q*, uses all of the directories (or symbolic links to directories) that begin with "q" as queue directories. Do not change the queue directory structure while sendmail is running. Queue runs create a separate process for running each queue unless the verbose flag (-v) is used on a nondaemon queue run. The new items are randomly assigned to a queue.

- The new queue file-naming system uses file names that are guaranteed to be unique for 60 years. This system allows queue IDs to be assigned without complex file-system locking and simplifies the movement of queued items between queues.
- Starting with version 8.12, only root can run the mail queue. For further details, refer to the changes that are described in the mailq(1) man page. For new task information, refer to "Administering the Queue Directories (Task Map)" on page 48.
- To accommodate envelope splitting, queue file names are now 15–characters long, rather than 14–characters long. File systems with a 14–character name limit are no longer supported.

For task information, refer to "Administering the Queue Directories (Task Map)" on page 48.

Changes for LDAP From Version 8.12 of sendmail

The following list describes changes in the use of the Lightweight Directory Access Protocol (LDAP) with sendmail.

- LDAPROUTE_EQUIVALENT() and LDAPROUTE_EQUIVALENT_FILE() permit you to specify equivalent host names, which are replaced by the masquerade domain name for LDAP routing lookups. For more information, refer to /etc/mail/cf/README.
- As noted in the Release Notes that are part of the sendmail distribution at ftp://ftp.sendmail.org, the LDAPX map has been renamed to LDAP. Use the following syntax for LDAP.

  ```
  Kldap ldap options
  ```
- This release supports the return of multiple values for a single LDAP lookup. Place the values to be returned in a comma-separated string with the -v option, as is shown.

  ```
  Kldap ldap -v"mail,more-mail"
  ```
- If no LDAP attributes are specified in an LDAP map declaration, all attributes that are found in the match are returned.
- This version of sendmail prevents commas in quoted key and value strings in the specifications of the LDAP alias file from dividing a single entry into multiple entries.
- This version of sendmail has a new option for LDAP maps. The option -V*separator* enables you to specify a separator so that a lookup can return both an attribute and a value that are separated by the relevant *separator*.
- In addition to using the %s token to parse an LDAP filter specification, you can use the new token, %0, to encode the key buffer. The %0 token applies a literal meaning to LDAP special characters.

The following example shows how these tokens differ for a “*” lookup.

TABLE 3–28 Comparison of Tokens

LDAP Map Specification	Specification Equivalent	Result
`-k"uid=%s"`	`-k"uid=*"`	Matches any record with a user attribute
`-k"uid=%0"`	`-k"uid=\2A"`	Matches a user with the name “*”

The following table describes additional LDAP map flags.

TABLE 3–29 Additional LDAP Map Flags

Flag	Description
`-1`	Requires a single match to be returned. If more than one match is returned, the results are the equivalent of no records being found.
`-r never|always|search|find`	Sets the LDAP alias dereference option.
`-Z size`	Limits the number of matches to return.

Change to the Built-In Mailer From Version 8.12 of `sendmail`

The old `[TCP]` built-in mailer is not available. Use the `P=[IPC]` built-in mailer instead. The interprocess communications (`[IPC]`) built-in mailer now enables delivery to a UNIX domain socket on systems that support it. You can use this mailer with `LMTP` delivery agents that listen on a named socket. An example mailer might resemble the following.

```
Mexecmail, P=[IPC], F=lsDFMmnqSXzA5@/:|, E=\r\n,
S=10, R=20/40, T=DNS/RFC822/X-Unix, A=FILE /system/volatile/lmtpd
```

The first mailer argument in the `[IPC]` mailer is now checked for a legitimate value. The following table provides possible values for the first mailer argument.

TABLE 3–30 Possible Values for the First Mailer Argument

Value	Description
`A=FILE`	Use for UNIX domain socket delivery
`A=TCP`	Use for TCP/IP connections
`A=IPC`	Is no longer available as a first mailer argument

Additional Rule Sets From Version 8.12 of sendmail

The following table lists the additional rule sets and describes what the rule sets do.

TABLE 3–31 New Rule Sets

Set	Description
check_eoh	Correlates information that is gathered between headers and checks for missing headers. This rule set is used with the macro storage map and is called after all of the headers have been collected.
check_etrn	Uses the ETRN command (as check_rcpt uses RCPT).
check_expn	Uses the EXPN command (as check_rcpt uses RCPT).
check_vrfy	Uses the VRFY command (as check_rcpt uses RCPT).

The following list describes additional rule set features.

- Numbered rule sets are also named, but the rule sets can still be accessed by their numbers.
- The H header configuration file command allows for a default rule set to be specified for header checks. This rule set is called only if the individual header has not been assigned its own rule set.
- Comments in rule sets (that is, text within parentheses) are not removed if the configuration file version is nine or greater. For example, the following rule matches the input token (1), but does not match the input token.

```
R$+ (1)        $@ 1
```

- sendmail accepts the SMTP RSET command even when it rejects commands because of TCP wrappers or the check_relay rule set.
- You receive a warning if you set the OperatorChars option multiple times. Also, do not set OperatorChars after the rule sets are defined.
- The name of the rule set, as well as its lines, are ignored if an invalid rule set is declared. The rule set lines are not added to S0.

Changes to Files From Version 8.12 of sendmail

Note the following changes.

- To support a read-only /usr file system, the contents of the /usr/lib/mail directory has been moved to the /etc/mail/cf directory. For details, refer to "Contents of the /etc/mail/cf Directory" on page 79. Note, however, that the shell scripts /usr/lib/mail/sh/check-hostname and /usr/lib/mail/sh/check-permissions are now in the /usr/sbin directory. See "Other Files Used for Mail Services" on page 82. For backward compatibility, symbolic links point to each file's new location.

- The new name for /usr/lib/mail/cf/main-v7sun.mc is /etc/mail/cf/cf/main.mc.
- The new name for /usr/lib/mail/cf/subsidiary-v7sun.mc is /etc/mail/cf/cf/subsidiary.mc.
- The helpfile is now located in /etc/mail/helpfile. The old name (/etc/mail/sendmail.hf) has a symbolic link that points to the new name.
- The trusted-users file is now located in /etc/mail/trusted-users. During an upgrade, if the old name (/etc/mail/sendmail.ct) is detected, but not the new name, a hard link from the old name to the new name is created. Otherwise, no change is made. The default content is root.
- The local-host-names file is now located in /etc/mail/local-host-names. During an upgrade, if the old name (/etc/mail/sendmail.cw) is detected, but not the new name, a hard link from the old name to the new name is created. Otherwise, no change is made. The default content is zero length.

sendmail Version 8.12 and IPv6 Addresses in Configuration

Starting with version 8.12 of sendmail, IPv6 addresses that are used in configuration should be prefixed with the IPv6: tag to identify the address properly. If you are not identifying an IPv6 address, a prefix tag is not used.

Index