Managing sendmail Services in Oracle® Solaris 11.4

Part No: E61008
November 2020

Managing sendmail Services in Oracle Solaris 11.4

Part No: E61008

Copyright © 2002, 2020, Oracle and/or its affiliates.

License Restrictions Warranty/Consequential Damages Disclaimer

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

Warranty Disclaimer

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

Restricted Rights Notice

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

Hazardous Applications Notice

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Trademark Notice

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

Third-Party Content, Products, and Services Disclaimer

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Pre-General Availability Draft Label and Publication Date

Pre-General Availability: 2020-01-15

Pre-General Availability Draft Documentation Notice

If this document is in public or private pre-General Availability status:

This documentation is in pre-General Availability status and is intended for demonstration and preliminary use only. It may not be specific to the hardware on which you are using the software. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to this documentation and will not be responsible for any loss, costs, or damages incurred due to the use of this documentation.

Oracle Confidential Label

ORACLE CONFIDENTIAL. For authorized use only. Do not distribute to third parties.

Revenue Recognition Notice

If this document is in private pre-General Availability status:

The information contained in this document is for informational sharing purposes only and should be considered in your capacity as a customer advisory board member or pursuant to your pre-General Availability trial agreement only. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle Master Agreement, Oracle License and Services Agreement, Oracle PartnerNetwork Agreement, Oracle distribution agreement, or other license agreement which has been executed by you and Oracle and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers that have purchased support 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.

Référence: E61008

Copyright © 2002, 2020, Oracle et/ou ses affiliés.

Restrictions de licence/Avis d'exclusion de responsabilité en cas de dommage indirect et/ou consécutif

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf stipulation expresse de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, accorder de licence, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Exonération de garantie

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Avis sur la limitation des droits

Si ce logiciel, ou la documentation qui l'accompagne, est livré sous licence au Gouvernement des Etats-Unis, ou à quiconque qui aurait souscrit la licence de ce logiciel pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique :

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

Avis sur les applications dangereuses

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer un risque de dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour des applications dangereuses.

Marques

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

Intel et Intel Inside sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. AMD, Epyc, et le logo AMD sont des marques ou des marques déposées d'Advanced Micro Devices. UNIX est une marque déposée de The Open Group.

Avis d'exclusion de responsabilité concernant les services, produits et contenu tiers

Ce logiciel ou matériel et la documentation qui l'accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers, sauf mention contraire stipulée dans un contrat entre vous et Oracle. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'accès à des contenus, produits ou services tiers, ou à leur utilisation, sauf mention contraire stipulée dans un contrat entre vous et Oracle.

Date de publication et mention de la version préliminaire de Disponibilité Générale ("Pre-GA")

Version préliminaire de Disponibilité Générale ("Pre-GA") : 15.01.2020

Avis sur la version préliminaire de Disponibilité Générale ("Pre-GA") de la documentation

Si ce document est fourni dans la Version préliminaire de Disponibilité Générale ("Pre-GA") à caractère public ou privé :

Cette documentation est fournie dans la Version préliminaire de Disponibilité Générale ("Pre-GA") et uniquement à des fins de démonstration et d'usage à titre préliminaire de la version finale. Celle-ci n'est pas toujours spécifique du matériel informatique sur lequel vous utilisez ce logiciel. Oracle Corporation et ses affiliés déclinent expressément toute responsabilité ou garantie expresse quant au contenu de cette documentation. Oracle Corporation et ses affiliés ne sauraient en aucun cas être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'utilisation de cette documentation.

Mention sur les informations confidentielles Oracle

INFORMATIONS CONFIDENTIELLES ORACLE. Destinées uniquement à un usage autorisé. Ne pas distribuer à des tiers.

Avis sur la reconnaissance du revenu

Si ce document est fourni dans la Version préliminaire de Disponibilité Générale ("Pre-GA") à caractère privé :

Les informations contenues dans ce document sont fournies à titre informatif uniquement et doivent être prises en compte en votre qualité de membre du customer advisory board ou conformément à votre contrat d'essai de Version préliminaire de Disponibilité Générale ("Pre-GA") uniquement. Ce document ne constitue en aucun cas un engagement à fournir des composants, du code ou des fonctionnalités et ne doit pas être retenu comme base d'une quelconque décision d'achat. Le développement, la commercialisation et la mise à disposition des fonctions ou fonctionnalités décrites restent à la seule discrétion d'Oracle.

Ce document contient des informations qui sont la propriété exclusive d'Oracle, qu'il s'agisse de la version électronique ou imprimée. Votre accès à ce contenu confidentiel et son utilisation sont soumis aux termes de vos contrats, Contrat-Cadre Oracle (OMA), Contrat de Licence et de Services Oracle (OLSA), Contrat Réseau Partenaires Oracle (OPN), contrat de distribution Oracle ou de tout autre contrat de licence en vigueur que vous avez signé et que vous vous engagez à respecter. Ce document et son contenu ne peuvent en aucun cas être communiqués, copiés, reproduits ou distribués à une personne extérieure à Oracle sans le consentement écrit d'Oracle. Ce document ne fait pas partie de votre contrat de licence. Par ailleurs, il ne peut être intégré à aucun accord contractuel avec Oracle ou ses filiales ou ses affiliés.

Accessibilité de la documentation

Pour plus d'informations sur l'engagement d'Oracle pour l'accessibilité de la documentation, visitez le site Web Oracle Accessibility Program, à l'adresse : `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Accès aux services de support Oracle

Les clients Oracle qui ont souscrit un contrat de support ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` ou le site `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` si vous êtes malentendant.

Contents

Using This Documentation 11

1 Introduction to `sendmail` 13
- About `sendmail` 13
  - Features of `sendmail` 13
  - Changes in This Release 15
  - Oracle Solaris Version of `sendmail` 15
- Components of Mail Services 17

2 Components of Mail Services 21
- Hardware Components for Mail Configurations 21
  - Mail Host 21
  - Mail Server 22
  - Mail Client 22
  - Mail Gateway 22
- Software Components for Mail Services 23
  - Mail Agents 24
  - Mailers and `sendmail` 24
  - Mailbox Files 25
  - Mail Aliases 26

3 Mail Services Directories and Files 29
- Essential Directories for Mail Services 29
  - Contents of the `/usr/bin` Directory 29
  - Contents of the `/etc/mail` Directory 30
  - Contents of the `/etc/mail/cf` Directory 31
  - Contents of the `/usr/lib` Directory 33

Essential Files for Mail Services 34
`sendmail` Configuration File 34
Mail Alias Files 35
`.forward` Files 38
`/etc/default/sendmail` File 40
Additional Directories and Files Used for Mail Services 41
How `sendmail` Works With Programs 42
Rerouting Mechanisms 43
Mail Addresses and Mail Routing 45
How `sendmail` Works With Name Services 45
`sendmail.cf` and Mail Domains 46
`sendmail` and Name Services 46
How `sendmail` Works With NIS and DNS 47
`vacation` Utility 48

4 Planning and Setting Up Mail Services 49
Planning Your Mail System 49
Local Mail Only 49
Local Mail and a Remote Connection 50
Setting Up Mail Services 51
▼ How to Verify Mailhost Entries in DNS 52

5 Administering Mail Services 53
Changing the `sendmail` Configuration 53
▼ How to Build a New `sendmail.cf` File 54
Setting Up a Virtual Host 55
Automatically Rebuilding Configuration Files 56
Using `sendmail` in Open Mode 56
Setting SMTP to use TLS 57
▼ How to Set SMTP to Use TLS 57
Managing SMTP Mail Delivery 63
▼ How to Manage Mail Delivery by Using an Alternate Configuration of `sendmail.cf` 63
Administering Mail Alias Files 64
▼ How to Set Up an NIS `mail.aliases` Map 64
▼ How to Set Up a Local Mail Alias File 65

▼ How to Create a File to Redirect Addresses 67
Managing the `postmaster` Alias 67
Administering the Mail Queue Directories 69
Displaying the Contents of the Mail Queue, `/var/spool/mqueue` 70
Forcing Mail Queue Processing in the `/var/spool/mqueue` Mail Queue 70
Running a Subset of the `/var/spool/mqueue` Mail Queue 71
▼ How to Move the `/var/spool/mqueue` Mail Queue 71
▼ How to Run the `/var/spool/omqueue` Old Mail Queue 72
Administering `.forward` Files 72
▼ How to Disable `.forward` Files 73
▼ How to Change the `.forward` File Search Path 74
Assigning Shell Access for `.forward` Permission 74
▼ How to Create and Populate the `/etc/shells` File 74
Switching Mail Transfer Agents 75
▼ How to Install and Run the `postfix` Service 75
▼ How to Revert to the `sendmail` Service 76

6 Troubleshooting Mail Services 77
Troubleshooting Procedures and Tips for Mail Services 77
Testing a New Mail Configuration 77
Verifying Mail Aliases 78
▼ How to Test the `sendmail` Rule Sets 78
Verifying Connections to Other Systems 79
Other Sources for Mail Diagnostic Information 80
Error Messages 80
Logging Error Messages 81
Resolving Error Messages 81

7 Changes in Versions of `sendmail` 85
Changes in Version 8.15 of `sendmail` 85
Changes in Version 8.14 of `sendmail` 86
Changes in Version 8.13 of `sendmail` 86
`FallBackSmartHost` Option 86
Support for Running SMTP With TLS in Version 8.13 of `sendmail` 87
Additional Command-Line Options in Version 8.13 of `sendmail` 92

- Additional and Revised Configuration File Options in Version 8.13 of `sendmail` 92
- Additional and Revised `FEATURE()` Declarations in Version 8.13 of `sendmail` 94

Changes in Version 8.12 of `sendmail` 94

- Support for TCP Wrappers in Version 8.12 of `sendmail` 95
- `submit.cf` Configuration File in Version 8.12 of `sendmail` 95
- Additional or Deprecated Command-Line Options in Version 8.12 of `sendmail` 97
- Additional Arguments for the `PidFile` and `ProcessTitlePrefix` Options in Version 8.12 of `sendmail` 98
- Additional Defined Macros in Version 8.12 of `sendmail` 99
- Additional Macros for the Configuration File in Version 8.12 of `sendmail` 100
- Additional `MAX` Macros in Version 8.12 of `sendmail` 100
- Additional and Revised `m4` Configuration Macros in Version 8.12 of `sendmail` 101
- Changes to the `FEATURE()` Declaration in Version 8.12 of `sendmail` 101
- Changes to the `MAILER()` Declaration in Version 8.12 of `sendmail` 105
- Additional Delivery Agent Flags in Version 8.12 of `sendmail` 105
- Additional Equates for Delivery Agents in Version 8.12 of `sendmail` 106
- Additional Queue Features in Version 8.12 of `sendmail` 107
- Changes for LDAP in Version 8.12 of `sendmail` 108
- Change to the Built-In Mailer in Version 8.12 of `sendmail` 109
- Additional Rule Sets in Version 8.12 of `sendmail` 109
- Changes to Files in Version 8.12 of `sendmail` 110
- `sendmail` Version 8.12 and IPv6 Addresses Used in Configuration 111

Index 113

Using This Documentation

- **Overview** – Describes how to configure the `sendmail` service to manage email.
- **Audience** – System administrators.
- **Required knowledge** – Basic and some advanced network administration skills.

Product Documentation Library

Documentation and resources for this product and related products are available at `http://www.oracle.com/pls/topic/lookup?ctx=E37838-01`.

Feedback

Provide feedback about this documentation at `http://www.oracle.com/goto/docfeedback`.

◆ ◆ ◆ CHAPTER 1

Introduction to `sendmail`

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 or set up a mail service on a new network or a subnet. Oracle Solaris uses a mail transfer agent called `sendmail` to handle mail services such as aliasing and forwarding.

This chapter describes the features of `sendmail`, and the differences between the Oracle Solaris version of `sendmail` and the generic Berkeley version. The chapter also provides an introduction to the various hardware and software components used for `sendmail`.

About `sendmail`

The `sendmail` service is a program that acts as a mail transfer agent. The `sendmail` program is responsible for receiving email messages from user agents and delivering them to local delivery agents. The Oracle Solaris operating system (OS) supplies standard configuration files for `sendmail`. The `sendmail` program uses the configuration files to provide aliasing, forwarding, automatic routing to network gateways, and flexible configuration.

Features of `sendmail`

The Oracle Solaris OS uses the `sendmail` program as a mail router. The Oracle Solaris version of `sendmail` has the following features:

- Delivers messages accurately without any data loss using communications protocols, such as TCP/IP. It implements an SMTP server, message queuing, and mailing lists. This program can be configured to handle complex environments including multiple networks. It also checks the contents of an address and its syntax to determine which mailer to use.
- Uses existing software for delivery whenever possible. You do not need to install additional software to perform a task. For example, the `sendmail` program works interactively and

manages both the mail-generating and the mail-sending program. When you send a mail, the mail-generating program calls `sendmail`, which routes the message to the correct mailers. Because some senders are network servers and some mailers are network clients, `sendmail` can be used as an Internet mail gateway. For more information, see "How `sendmail` Works With Programs" on page 42.

- Uses configuration files to control mail configuration. You do not need to compile the configuration information in the code of the `sendmail` program. The program enables you to use different configuration files for different environments. For example, the Oracle Solaris OS provides a default configuration file for the `sendmail` program. Create a backup of the default configuration file before you modify the file according to your environment. The modified configuration file is tested and then used. The `sendmail` program enables you to create many such configuration files for many environments without having to lose the default configuration file. For more information, see "`sendmail` Configuration File" on page 34.
- Can be configured to automatically rebuild the `sendmail.cf` and the `submit.mc` configuration files. For more information, see "Automatically Rebuilding Configuration Files" on page 56.

The `sendmail` program controls name interpretation by using a pattern-matching system that works with the following naming conventions:

- Domain-based naming convention. The domain technique distinguishes physical naming from logical naming
- Improvised techniques, such as providing network names that appear local to hosts on other networks
- Arbitrary (older) naming syntaxes
- Disparate naming schemes

The `sendmail` program controls email messages that users send in the following ways:

- Evaluates the recipients' addresses
- Chooses an appropriate delivery program
- Rewrites the addresses in a format that the delivery agent can handle
- Reformats the mail headers as required
- Passes the transformed message to the mail program for delivery

For more information, see "Rerouting Mechanisms" on page 43 and "`sendmail` Configuration File" on page 34.

Changes in This Release

The following changes have been made in the Oracle Solaris 11.4 release:

- The default version of `sendmail` was updated to 8.15.2.
- Mailman Mailing List Manager is integrated to manage email discussions and e-newsletter lists.
- Enables you to configure `ServerSSLOptions` and `ClientSSLOptions`.
- Enables you to switch between mail transfer agents such as Postfix.

Oracle Solaris Version of `sendmail`

This section 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`

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
```

This command does not list the flags that are specific to Oracle Solaris.

If your configuration requires other flags, you need to download the source and recompile the binary. For more information, see Compiling Sendmail (`https://web.archive.org/web/20161025083736/http://www.sendmail.com:80/sm/open_source/tips/compiling_sendmail/`).

The following flags are used when compiling the `sendmail` binary.

`SOLARIS=21100`

Support for the Oracle Solaris 11 release.

`MILTER`

Support for the Mail Filter API. In version 8.13 and later versions of `sendmail`, this flag is enabled by default.

`NETINET6`

Support for IPv6. This flag has been moved from `conf.h` to `Makefile`.

`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`.

The following OS-specific compile flags are used when building the `sendmail` library.

`SUN_EXTENSIONS`

Support for various Oracle Solaris added extensions.

`SUN_INIT_DOMAIN`

For backward compatibility, support for the use of NIS domain names to fully qualify the local host name.

`SUN_SIMPLIFIED_LDAP`

Support for a simplified LDAP API, which is specific to Oracle Solaris.

`VENDOR_DEFAULT=VENDOR_SUN`

Selects Oracle Solaris as the default vendor.

`STARTTLS`

Transaction Level Security (RFC 2487)

The following generic flag is not used to compile the version of `sendmail`.

`SASL`

Simple Authentication and Security Layer (RFC 2554)

Alternative `sendmail` Commands

Oracle Solaris does not include all the command synonyms that are provided in the generic release from the Open Source sendmail organization. The following table provides information about the command aliases.

TABLE 1 Alternative `sendmail` Commands

Alternate Name	Available in Oracle Solaris?	Options With `sendmail`
`hoststat`	No	`sendmail -bh`

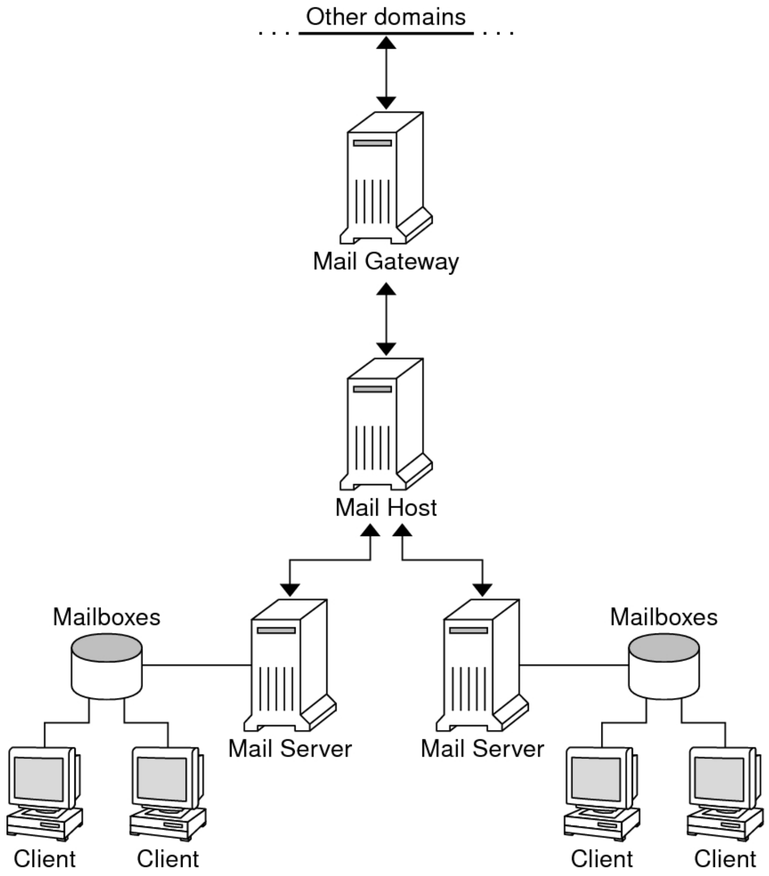
Alternate Name	Available in Oracle Solaris?	Options With `sendmail`
`mailq`	Yes	`sendmail -bp`
`newaliases`	Yes	`sendmail -bi`
`purgestat`	No	`sendmail -bH`
`smtpd`	No	`sendmail -bd`

Components of Mail Services

Establishing a mail service requires many hardware and software components.

The following figure shows an electronic mail configuration with related hardware components.

FIGURE 1 Electronic Mail Configuration


Other domains
Mail Gateway
Mail Host
Mailboxes
Mail Server
Mail Server
Mailboxes
Client
Client
Client
Client

A mail configuration requires the following hardware components, which can either be combined on the same system or provided on separate systems:

- Mail host – A system that is configured to handle email addresses that are difficult to resolve
- Mail server – A minimum of one system that is configured to hold one or more mailboxes
- Mail clients – Systems that access mail from a mail server
- Mail gateway – A system used to communicate with the networks outside your domain

For more information, see "Hardware Components for Mail Configurations" on page 21.

A mail service requires the following software components to function:

- `.forward` files – Files that you can set up in a your home directory to redirect mail or send mail to a program automatically
- `mailbox` – A file on a mail server that is the final destination for email messages
- Mail addresses – Addresses that contains the name of the recipient and the system to which a mail message is delivered
- Mail aliases – Alternate names that are used in a mail address
- Mail queue – A collection of mail messages that need to be processed by the mail server
- Postmaster – A special mail alias that is used to report problems and ask questions about the mail service
- `sendmail` configuration file – A file that contains all the information necessary for mail routing

For more information, see “Software Components for Mail Services” on page 23.

◆ ◆ ◆ CHAPTER 2

Components of Mail Services

Establishing a mail service requires many hardware and software components. This chapter describes the hardware and software components that are required to set up and administer mail services.

This chapter covers the following topics:

- "Hardware Components for Mail Configurations" on page 21
- "Software Components for Mail Services" on page 23

Hardware Components for Mail Configurations

A mail configuration requires certain hardware components to be present either on the same system or on separate systems: a mail host, a mail server, a mail client, and a mail gateway.

Mail Host

A *mail host* is the primary component on your network to which other systems forward mail that cannot be delivered. You can 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. Alternatively, you can add the word `mailhost` to the `hosts` file in the name service.

A good candidate for a mail host is a system that is configured as a router on your network and connects to the Internet. For more information, see "Configuring an IPv4 Router" in *Configuring an Oracle Solaris 11.4 System as a Router or a Load Balancer*.

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 email for the configuration by designating the standalone system as the mail host of a single-system network.

Mail Server

A *mailbox* is a file that stores email for a particular user. Mail is delivered to the user's mailbox where the mailbox is located either on a local machine or on a remote server. A *mail server* maintains user mailboxes using the `/var/mail` directory.

The mail server routes all mail from a client. When a client sends a mail message, the mail server puts the message in the queue for delivery. After the message is in the queue, the sender can reboot or turn off the client without losing it. If the recipient responds, the response goes to the sender'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 a configuration that uses NFS software can mount the `/var/mail` directory provided the user has `root` access. Otherwise, the user can use the automounter to mount the `/var/mail` directory. If NFS support is not available, the user can log in to the server to read mail.

If users on your network typically send messages with attachments, such as audio files or files from desktop publishing systems, you might 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 creating backups. 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 advantage of creating backups usually makes the risk worthwhile.

Mail Client

A mail client uses mail services and has a mailbox that is present 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.

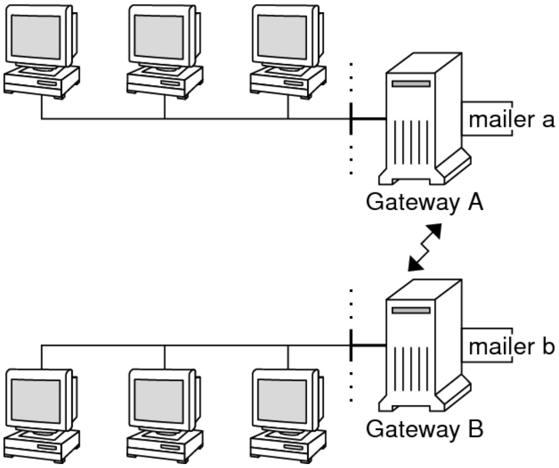
Mail Gateway

The *mail gateway* handles connections between networks. Networks can use either different communication protocols or the same communication protocol. For example, a mail gateway might connect a TCP/IP network to a network that uses the Systems Network Architecture (SNA) protocol suite.

A mail gateway connecting two networks that use the same protocol or mailer is easy to set up. The mail gateway handles mail in your domain sent to recipients that the `sendmail` program cannot find. If a mail gateway exists, `sendmail` uses the gateway to send and receive mail outside your domain.

The following figure shows a mail gateway between two networks that use unmatched mailers.

FIGURE 2 Gateway Between Different Communication Protocols



To support this configuration, you must customize the `sendmail.cf` file on the mail gateway system.

If you have a machine that connects 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.

Software Components for Mail Services

Each mail service uses several software components. This section describes the following software components:

- "Mail Agents" on page 24

- "Mailers and `sendmail`" on page 24
- "Mailbox Files" on page 25
- "Mail Aliases" on page 26

Mail Agents

The mail service uses the following agents:

Mail user agent – The program that acts as the interface between the user and the `sendmail` mail transfer agent. The Oracle Solaris OS supplies the `/usr/bin/mail` and `/usr/bin/mailx` mail user agents.

Mail transfer agent – Responsible for the routing of mail messages and the resolution of mail addresses. This agent is also known as a mail transport agent. The `sendmail` program is the default mail transfer agent for the Oracle Solaris OS. The mail transfer agent performs the following functions:

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

Local delivery agent – A program that implements a mail delivery protocol. The Oracle Solaris OS provides the `mail.local` and `procmail` local delivery agents.

Mailers and `sendmail`

Mailer is a `sendmail`-specific term. A mailer is used by the `sendmail` program 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. The sendmail program uses the Simple Mail Transfer Protocol (SMTP) mailer. The SMTP protocol defines the following 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

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. This practice enables the user agent to pull mail from the mail spool and store it 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 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 mail handling.

The `/etc/mail/aliases` file and name services such as NIS provide mechanisms for creating aliases for email addresses. Users do not need to know the precise local name of their mailboxes.

The following list shows some common naming conventions for special-purpose mailboxes:

username	User names are frequently the same as mailbox names.
Firstname`.`*Lastname* *Firstname*`_`*Lastname* *Firstinitial*`.`*Lastname* *Firstinitial*`_`*Lastname*	User names are full names with a dot (or an underscore) that separates the first and last names. User names can also be identified by the 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 has a `postmaster` mailbox.
`MAILER-DAEMON`	The `sendmail` program 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 redirects mail to the user who maintains the distribution list.

`owner-`*aliasname*	Names that begin with `owner-` are administrative addresses for distribution lists. This address redirects mail to the user who handles mail errors.
`owner-owner`	This alias is used when no `owner-`*aliasname* alias exists to receive errors that are returned. This address redirects mail to the user who handles mail errors. This address 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. Many people use the `%` convention, but the 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 was changed so that mail sent to an alias looks like the mail came from the alias owner when the mail is delivered. This change enables mail errors to be sent to the alias owner, rather than being returned to the sender. 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: tamiko, leslie
```

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 the mail sent to the `mygroup` alias, the envelope sender changes to `mygroup-request`.

Mail Aliases

An *alias* is an alternate name that can be used to assign a mailbox location or to define mailing lists. For more information, see “Administering Mail Alias Files” on page 64 and “Mail Alias Files” on page 35.

For large sites, the mail alias typically defines the location of a mailbox. Providing a mail alias is similar to 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 the mail is to be delivered within the building. So, the possibility of an error increases. For example, if two people named Kevin Smith are in the same building, only one of them gets the mail. To correct the problem, each Kevin Smith should have a room number added to his address.

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

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

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 `tamiko`:

```
tamiko: tamiko.smith
tamikos: tamiko.smith
tamiko.smith: tamiko@mars
```

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

```
leslie: leslie@north
```

When you create mail lists that include users outside your domain, create the alias with the user name and the domain name. For example, you have a user named `tamiko` on system `privet`, in domain `example.com`. Create the alias as `tamiko@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.

You can use the following methods for creating and administering mail alias files:

- 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.
- Administer aliases by using the NIS name service to maintain a global `aliases` database depending on the configuration of your mail services. Otherwise, you can update all the local `/etc/mail/aliases` files to keep the aliases synchronized.
- Create and use aliases either in your local `~/.mailrc` file, which only you can use, or in your local `/etc/mail/aliases` file, which anyone can use. NIS alias files cannot be created or administered by a regular user.

◆ ◆ ◆ CHAPTER 3

Mail Services Directories and Files

Mail services include many programs and daemons that interact with each other. This chapter covers the following topics:

- "Essential Directories for Mail Services" on page 29
- "Essential Files for Mail Services" on page 34
- "Additional Directories and Files Used for Mail Services" on page 41
- "How `sendmail` Works With Programs" on page 42
- "`vacation` Utility" on page 48

Essential Directories for Mail Services

The directories that are used to manage and administer mail services are:

- `/usr/bin`
- `/etc/mail`
- `/etc/mail/cf`
- `/usr/lib`

Contents of the `/usr/bin` Directory

The contents of the `/usr/bin` directory is used for mail services:

`mail`	A user agent.
`mailq`	A program that lists the content of the mail queue.

`mailstats`	A program that is used to read mail statistics that are stored in the `/etc/mail/statistics` file (if present).
`mailx`	A user agent.
`mconnect`	A program that connects to the mailer for address verification and debugging.
`praliases`	A command to "uncompile" the alias database. For more information, see the `praliases(1)` man page.
`rmail`	A symbolic link to the `/usr/bin/mail` command that is often used to permit only the sending of mail.
`vacation`	A command to set up an automatic reply to mail.

Contents of the `/etc/mail` Directory

The contents of the `/etc/mail` directory are:

`Mail.rc`	Default settings for the `mailx` user agent.
`aliases`	Mail-forwarding information.
`aliases.db`	Default binary form of mail-forwarding information that is created by running `newaliases`.
`aliases.dir`	Binary form of mail-forwarding information that is created by running `newaliases`. Is not used by default.
`aliases.pag`	Binary form of mail-forwarding information that is created by running `newaliases`. Is not used by default.
`mailx.rc`	Default settings for the `mailx` user agent.
`main.cf`	Symbolic link from the sample configuration file for main systems. The file links to `sendmail.cf` and is provided for backward compatibility. This file is not needed in version 8.12 or later versions of `sendmail`.
`relay-domains`	List of all domains for which relaying is allowed. By default, relaying is allowed only for the local domain.

`sendmail.cf`	Configuration file for mail routing.
`submit.cf`	New configuration file for the mail submission program. For more information, see “`submit.cf` Configuration File in Version 8.12 of `sendmail`” on page 95.
`local-host-names`	Optional file that you create if the number of aliases for the mail host is too long.
`helpfile`	Help file that is used by the SMTP `HELP` command.
`sendmail.pid`	File that lists the PID of the listening daemon and is present in `/system/volatile`.
`statistics`	The `sendmail` program statistics file. The `sendmail` program logs the amount of traffic through each mailer. Previously, this file was called `sendmail.st`.
`subsidiary.cf`	Symbolic link from the sample configuration file for subsidiary systems. Points to `sendmail.cf` and is provided for backward compatibility. This file is not needed in version 8.13 of `sendmail`.
`trusted-users`	File that lists the users (one user per line) who are 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

The `/etc/mail` directory consists of a subdirectory `cf` containing all the necessary files required to build a `sendmail.cf` file.

To support a read-only `/usr` file system, the content of the `/usr/lib/mail` directory was moved to the `/etc/mail/cf` directory in Oracle Solaris 10. Note, however, that the shell scripts `/usr/lib/mail/sh/check-hostname` and `/usr/lib/mail/sh/check-permissions` were moved to `/usr/sbin` directory. For more information, see “Additional Directories and Files Used for Mail Services” on page 41. For backward compatibility, symbolic links point to each file's new location.

The contents of the `cf` subdirectory are as follows:

`README`	Describes the configuration files.
`cf/main.cf`	Points to `cf/sendmail.cf`. This file used to be the main configuration file.
`cf/main.mc`	Points to `cf/sendmail.mc`. This file used to create the main configuration file.
`cf/Makefile`	File that provides rules for building new configuration files.
`cf/submit.cf`	Configuration file for the mail submission program, which is used to submit messages.
`cf/submit.mc`	File that is used to build the `submit.cf` file. This file defines `m4` macros for the mail submission program.
`cf/sendmail.cf`	Main configuration file for `sendmail`.
`cf/sendmail.mc`	File that contains the `m4` macros that are used to generate the `sendmail.cf` file.
`cf/subsidiary.cf`	Symbolic link that points 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 that points to to `cf/sendmail.mc`. This file used to contain the `m4` macros that were used to generate the `subsidiary.cf` file.
`domain`	Directory that provides site-dependent subdomain descriptions.
`domain/generic.m4`	Generic domain file from Berkeley Software Distribution.
`domain/solaris-antispam.m4`	Domain file that makes the current version of `sendmail` function like the previous versions of `sendmail` by making a few changes. However, note that relaying is disabled completely. Sender's addresses with no host name and unresolvable domains are rejected.
`domain/solaris-generic.m4`	Default domain file that makes the current version of `sendmail` function like the previous versions of `sendmail` by making a few changes.

`feature`	Directory that contains definitions of specific features for particular hosts. For a full description of the features, see the `README` file in the `/etc/mail/cf` directory.
`m4`	Directory that contains site-independent `include` files.
`mailer`	Directory that contains definitions of mailers, which include `local`, `smtp`, and `uucp`.
`ostype`	Directory that contains descriptions of various OS environments.
`ostype/solaris2.m4`	File that defines the default local mailer as `mail.local`.
`ostype/solaris2.ml.m4`	File that defines the default local mailer as `mail.local`.
`ostype/solaris2.pre5.m4`	File that defines the local mailer as `mail`.
`ostype/solaris8.m4`	File that defines the local mailer as `mail.local` (in LMTP mode), enables IPv6, and specifies `/var/run` as the directory for the `sendmail.pid` file.
`ostype/solaris8.m4`	File that defines the local mailer as `mail.local` (in LMTP mode), enables IPv6, and specifies `/system/volatile` as the directory for the `sendmail.pid` file.

Contents of the `/usr/lib` Directory

The files and symbolic link in the `/usr/lib` directory are used for mail services:

`mail.local`	Mailer that delivers mail to mailboxes.
`sendmail`	Routing program, also known as the mail transfer agent.
`smrsh`	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. For more information, see the `smrsh`(8) man page. To enable this program, include the `m4` command, ``FEATURE(`smrsh')`` in your `mc` file.

`mail`	Symbolic link. Points to the`/etc/mail/cf` directory. For more information, see "Contents of the `/etc/mail/cf` Directory" on page 31.

Essential Files for Mail Services

This section describes the following files that are required to manage and administer mail services:

- "`sendmail` Configuration File" on page 34
- "Mail Alias Files" on page 35
- "`.forward` Files" on page 38
- "`/etc/default/sendmail` File" on page 40

`sendmail` Configuration File

The `sendmail` configuration file controls the way the `sendmail` program 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 OS provides the following default configuration files in the `/etc/mail` directory:

- `sendmail.cf` – A configuration file used to run `sendmail` in daemon mode. For more information, see the `sendmail`(8) man page.
- `submit.cf` – A configuration file used to run `sendmail` in mail-submission program mode. For more information, see "`submit.cf` Configuration File in Version 8.12 of `sendmail`" on page 95.

When setting up a mail server, you do not need to set up or edit the default configuration file.

When you set up a mail host, you need to set the relay mailer and relay host parameters that are needed for your mail configuration. For more information, see "Changing the `sendmail` Configuration" on page 53. Note that with `sendmail` version 8.13, you no longer need the `main.cf` file.

You can change the following configuration parameters:

- 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. For more information, see "Additional Queue Features in Version 8.12 of `sendmail`" on page 107 and "Administering the Mail Queue Directories" on page 69.
- 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 logs different kinds of problems.

Versions of the Configuration File

The `sendmail` program 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 to be either Berkeley or Sun. Both are valid vendor options. If a version level is specified without the vendor, then `Sun` is chosen as the default vendor.

The following versions of the configuration file are used for `sendmail`:

- `V7/Sun` – Version used for version 8.8 of `sendmail`.
- `V8/Sun` – Version used for version 8.9 of `sendmail`.
- `V9/Sun` – Version used for versions 8.10 and 8.11 of `sendmail`.
- `V10/Sun` – Version used for version 8.12, 8.13, and 8.14 versions of `sendmail`. Version 8.13 of `sendmail` was used by default in the Oracle Solaris 10 release. Version 8.14 of `sendmail` is used by default for the Oracle Solaris 11 release.

For more information, see "Changing the `sendmail` Configuration" on page 53.

Mail Alias Files

The method of maintaining aliases depends on who uses the alias and who requires the ability to change the alias. Each type of alias has unique format requirements.

You can use any of the following items to maintain aliases:

- `.mailrc Aliases` file

- `/etc/mail/aliases` file
- NIS aliases Map

For more information, see "Administering Mail Alias Files" on page 64.

`.mailrc` Aliases File

Aliases listed in a `.mailrc` file are accessible only if you are the owner of the file. This restriction enables you to establish an alias file that only you control and use as the owner. Aliases in a `.mailrc` file adhere to the following format:

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

aliasname	The name you use when sending mail
value	A valid email address

For example, an error occurs if the personal alias for `jan` does not match the email address for `jan` in the name service. Mail is routed to the wrong person when people reply to mail generated by you. 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 lists in a local `/etc/mail/aliases` file adhere to the following format:

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

aliasname	The name you use when sending mail to this alias
value	A valid email address

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

The aliases in the `/etc/mail/aliases` file are stored in text form. After you edit the `/etc/mail/aliases` file, you need to run the `newaliases` program to recompile the database and make the aliases available in binary form to the `sendmail` program. For more information, see "How to Set Up a Local Mail Alias File" on page 65.

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 `tamiko` having a mailbox on the system `saturn` would have the following entry in the `/etc/mail/aliases` file.

```
tamiko: tamiko@saturn
```

You must create an administrative account for each mail server. You can create an administrative 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.

Only the `root` user can edit this file but another option is to create the following entry:

```
aliasname: :include:/path/alias-file
```

aliasname	The name you use when sending mail
/path/alias-file	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 more information, see "Administering Mail Alias Files" on page 64.

NIS `aliases` Map

The NIS `aliases` map contains entries for all mail clients. In general, only the `root` user on the NIS master can change these entries. This type of alias is not 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:

aliasname: *aliasname@host*

aliasname	The name you use when sending mail
host	The host name of the server that contains an `/etc/mail/alias` file

All users in a local domain can use the entries that are in the NIS `aliases` map because the `sendmail` program can use the NIS `aliases` map instead of the local `/etc/mail/aliases` files to determine mailing addresses. For more information, see the `nsswitch.conf`(5) man page.

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

aliasname: *value,value,value* . . .

aliasname	The name you use when sending mail
value	A valid email address

For more information, see "How to Set Up an NIS `mail.aliases` Map" on page 64.

`.forward` Files

You create `.forward` files in your home directory that `sendmail`, along with other programs, use to redirect mail or send mail.

This section covers the following topics:

- "Forwarding Situations to Avoid" on page 38
- "Controls for `.forward` files" on page 39
- "`.forward.`*hostname* File" on page 39
- "`.forward+`*detail* File" on page 40

For more information, see "Administering `.forward` Files" on page 72.

Forwarding Situations to Avoid

You can easily avoid or fix the following situations:

- If mail is not being delivered to the expected address, check the user's `.forward` file. For example, the user might have stored 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 an infinite loop of sending mail.

- To avoid security problems, never store `.forward` files in the `root` and `bin` accounts. If necessary, forward the mail by using the `aliases` file instead.

Controls for `.forward` files

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

- The owner of the `.forward` file is the only person with write permissions on the file. This restriction increases security.
- The `root` user should own and have write permissions for the paths that lead to the home directory. For example, if a `.forward` file is in `/export/home/terry`, `/export` and `/export/home`, then the `.forward` file is owned by the `root` user only. The `root` user also has write permissions for the file.
- The user is the only person with write permissions for the actual home directory.
- The `.forward` file cannot be a symbolic link and this file cannot have more than one hard link.

`.forward.`*hostname* File

You 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 address 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 delivers back to the address `sandy@example.com` but the user is able to see a change in the `To:` mail header.

`/etc/default/sendmail` File

The `/etc/default/sendmail` file is used to store startup options for `sendmail` so that the options are not removed when a host is upgraded. The file can contain the following variables:

`CLIENTOPTIONS="`*string*`"`

Selects additional options to be used with the client daemon. The option looks in the client-only queue (`/var/spool/clientmqueue`) and acts as a client queue runner. No syntax checking is done, so be careful while making changes to this variable.

`CLIENTQUEUEINTERVAL=`*#*

Sets the time interval for mail queue runs. Similar to the `QUEUEINTERVAL` option except that 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, the messages then 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. These intervals are periodic. The server can immediately deliver the portion of its queue which goes to the specified hosts. For more information, see the `etrn`(8) man page.

`MODE=-bd`

Chooses the mode to start `sendmail` by using the `-bd` option or leaving it undefined. For more information, see the `sendmail`(8) man page.

`OPTIONS=`*string*

Selects additional options to be used with the master daemon. No syntax checking is done, so be careful while making changes to this variable.

`QUEUEINTERVAL=`*#*

Sets the interval for mail queue runs on the master daemon. The positive integer *#* 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.

Additional Directories and Files Used for Mail Services

The following list describes the directories and files associated with mail services that have not previously been discussed:

`/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 lists the PID of the listening daemon.

`/usr/sbin/check-permissions`

File checks the permissions of the `:include:` aliases, `.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`

Mail notification daemon.

`/usr/sbin/makemap`

File builds binary forms of keyed maps.

`/usr/sbin/newaliases`

Symbolic link points to `/usr/lib/sendmail`. Used to create the binary form of the alias database. Previously in `/usr/bin`.

`/usr/sbin/syslogd`

Error message logger used by `sendmail`.

`/usr/sbin/etrn`

Perl script that starts the client-side remote mail queue.

`/var/mail/`*mailbox1*
`/var/mail/`*mailbox2*

Mailboxes for delivered mail.

`/var/spool/clientmqueue`

Storage directory for mail that is delivered by the client daemon.

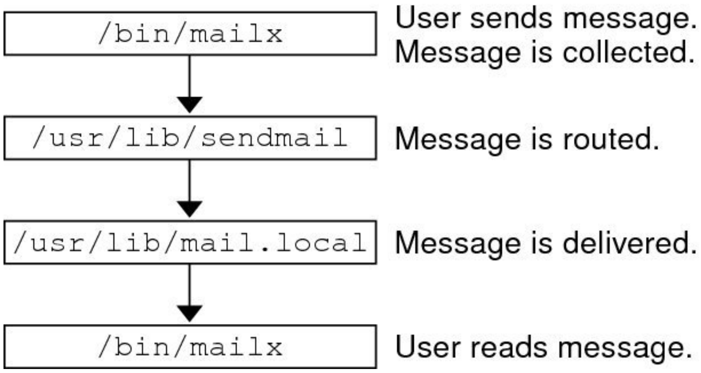
`/var/spool/mqueue`

Storage directory for mail that is delivered by the master daemon.

How `sendmail` Works With Programs

Mail services work in combination with different programs. The following figure shows how the programs work together.

FIGURE 3 How `sendmail` Works


/bin/mailx
User sends message.
Message is collected.
/usr/lib/sendmail
Message is routed.
/usr/lib/mail.local
Message is delivered.
/bin/mailx
User reads message.

The `sendmail` program performs the following steps when sending a mail message:

1. The message is sent by using a program such as `mailx`. For more information, see the `mailx`(1) man page.
2. The message is collected by the program that generates the message, and the message is then 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 the route the message must follow to get to the 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.

Rerouting Mechanisms

The `sendmail` program supports three mechanisms for mail rerouting: aliasing, forwarding, and inclusion.

The rerouting mechanism that you choose can affect the level of administration that is required.

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

  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 be able to administer this file.

  Some considerations when using a server alias file:

  - Rerouting can be managed by anyone who becomes the `root` user on the designated server. Standard users cannot administer the file.
  - Server aliasing should create little or no lag time when the rerouting change is propagated.
  - The change affects only the local server, which might be acceptable if most mail is sent to one server. However, if you need to propagate this change to many mail servers, use a name service.

  For more information, see "Mail Alias Files" on page 35 and "Administering Mail Alias Files" on page 64.
- Forwarding enables users to create `.forward` files to reroute their incoming mail to another mailbox, a different mailer, or another mail host.

  For more information, see "`.forward` Files" on page 38 and "Administering `.forward` Files" on page 72.
- *Inclusion*

  Inclusion involves users maintaining 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, see "`/etc/mail/aliases` File" on page 36 and "Administering Mail Alias Files" on page 64.

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. For more information, see the `nsswitch.conf(5)` man page.

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 the same on most hosts.

You can set up a client system to receive mail locally by running `sendmail` in local mode. Local mode is the default mode for all mail servers and some clients. In local mode, a mail message is routed as follows:

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. If a new value is found, it replaces the old value. The 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 the mail domain, deliver the mail to that host.
4. If the address does not include a host in the mail 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.

Note - The list assumes that you are using the default rule set in the `sendmail.cf` file.

How `sendmail` Works With Name Services

`sendmail` works with domain names and name services to deliver and route mail. This section covers the following topics:

- "`sendmail.cf` and Mail Domains" on page 46
- "`sendmail` and Name Services" on page 46
- "How `sendmail` Works With NIS and DNS" on page 47

For more information, see "How to Verify Mailhost Entries in DNS" on page 52 and "Administering Mail Alias Files" on page 64.

`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 conditions:

- 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. The mail host can possibly be an authorized host

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

`sendmail` and Name Services

The `sendmail` program 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 and then describes the various requirements.

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 values:

```
A.B.C.D
B.C.D
C.D
D
```

The mail domain name is identical to the name service domain when it is first established. As the network grows, the name service domain can be divided to make the name service more manageable. However, the mail domain remains undivided to provide consistent aliasing.

Requirements for Name Services

The `sendmail` program requires that the following host table or map in a name service be set up to support three types of `gethostbyname()` queries:

- `mailhost` – Some name service configurations satisfy this requirement automatically.
- Full host name – Many name service configurations satisfy this requirement. For example, `smith.admin.example.com` is the full host name.
- Short host name – The `sendmail` program 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. For example, if `smith.admin.example.com` is the full host name, then `smith` is the short host name.

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

Note the following additional requirements for 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.example.com)` should return the same result as `gethostbyname(smith)`, if both functions are called from the mail domain `admin.example.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.example.com` is given, `gethostbyname(smith)` should return the same result when the call originates from either the `sales.admin.example.com` domain or the `eng.admin.example.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, see the `gethostbyname`(3C) man page.

How `sendmail` Works With NIS and DNS

The `sendmail` program works with NIS and DNS as follows:

- 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, `sales.admin.example.com` becomes `admin.example.com`.

- Mail host name – You must have a `mailhost` entry in the NIS host map. When the DNS forwarding feature is turned on, queries that NIS cannot resolve are forwarded to DNS. Hence, you do not need a `mailhost` entry in the NIS host map.
- 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 `sales.admin.example.com` domain should be the same as the host map in the `eng.admin.example.com`. Otherwise, one address might work in one NIS domain but fail in the other NIS domain.

For more information, see "How to Verify Mailhost Entries in DNS" on page 52 and "Administering Mail Alias Files" on page 64.

`vacation` Utility

The `vacation` utility enables users to automatically send a message in response to received messages. Users can also specify which incoming messages receive autogenerated replies. With this enhancement, the user can avoid sharing confidential or contact information with unknown people, often spammers.

You can create a `.vacation.filter` file that contains a list of domains or email addresses and place it in your 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 the following entries:

```
example.com
example.org
leslie@west.example.com
jan@sales.example.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 an email address entry, the match must be exact, except for case. The matching is case insensitive. For a sender's email address to match a domain entry, the sender's address must contain the listed domain. For example, both `jan@dept.example.com` and `tamiko@example.com` would be a match for a domain entry of `example.com`.

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

◆ ◆ ◆

CHAPTER 4

Planning and Setting Up Mail Services

This chapter provides information about how to perform tasks such as planning and setting up mail services. It covers the following topics:

- "Planning Your Mail System" on page 49
- "Setting Up Mail Services" on page 51

Planning Your Mail System

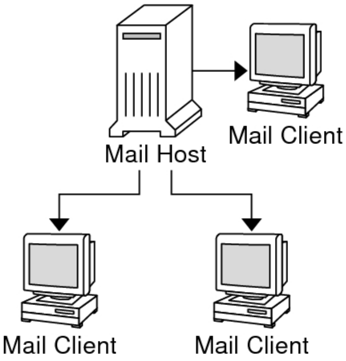
Consider the following during your planning process:

- Determine the type of mail configuration that meets your requirements. Check whether you need to set up a new mail system or if you need to expand an existing one: either local mail only or local mail with a remote connection. For more information, see "Local Mail Only" on page 49 and "Local Mail and a Remote Connection" on page 50.
- Select the systems that are to act as mail servers, mail hosts, and mail gateways.
- Make a list of all the mail clients that you are providing service to and include the location of their mailboxes. This list can help you when you are ready to create mail aliases for your users.
- Choose how to update aliases and forward mail messages. You can set up an `aliases` mailbox for users to send requests for mail forwarding. Users can use this mailbox to send requests for changes to their default mail alias. If your system uses NIS, you can administer mail forwarding without involving the users. For more information, see "Administering Mail Alias Files" on page 64, "Administering `.forward` Files" on page 72.

Local Mail Only

The following figure shows a simple mail configuration consisting of three clients connected to one mail host.

FIGURE 4 Local Mail Configuration



The configuration shown in the figure is simple and easy to set up. Mail is 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. You need the following components to set up this mail configuration:

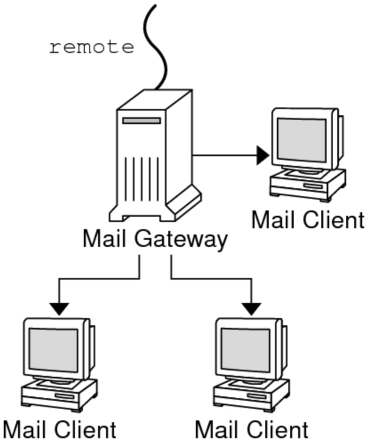
- The default `/etc/mail/sendmail.cf` file on each mail client system.
- A server that is designated as the mail host. If you are running NIS, you can assign this designation by adding `mailhost.`*domain-name* to the `/etc/hosts` file on the mail host. If you are running another name service such as DNS or LDAP, you must provide additional information in the `/etc/hosts` file.
- If you are using a name service other than NIS, you need to have the same `/etc/mail/aliases` files on all systems that have a local mailbox.
- You need enough disk space in `/var/mail` directory on each mail client system to hold the mailboxes.

For more information, see “Setting Up Mail Services” on page 51.

Local Mail and a Remote Connection

The following figure shows a common mail configuration for small networks.

FIGURE 5 Local Mail Configuration With a Remote Connection



The system includes the mail server, the mail host, and the mail gateway providing the remote connection. Mail is distributed by using the `/etc/mail/aliases` files on the mail gateway. No name service is required. In the configuration, the mail clients mount their mail files from `/var/mail` onto the mail host. You need the following components to set up this mail configuration:

- The default `/etc/mail/sendmail.cf` file on each mail client system
- A server designated as the mail host. If you are running NIS, you can assign 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.
- If you are using a name service other than NIS, you need matching `/etc/mail/aliases` files on all systems that have a local mailbox
- Enough disk space in `/var/mail` on the mail server to hold the client mailboxes

For more information, see "Setting Up Mail Services" on page 51.

Setting Up Mail Services

You can 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 components for local mail configuration – a mail host and mail clients. For communication with networks outside your domain, you also need a mail gateway and mail servers. For a graphical illustration of configurations for communication with networks outside your domain, see Figure 1, “Electronic Mail Configuration,” on page 18 in “Components of Mail Services” on page 17 or Figure 5, “Local Mail Configuration With a Remote Connection,” on page 51 in “Local Mail and a Remote Connection” on page 50.

There are no special steps 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. For mail to be delivered, the user should also have a local home directory for checking the `~/.forward` file. For this reason, home directory servers are often set up as the mail server. For more information, see “Hardware Components for Mail Configurations” on page 21.

You can combine components on the same system or provide components on separate systems. For example, if your mail host and mail server functions are on the same system, follow the instructions in this section to set up that system as the mail host. Then, follow the instructions in this section to set up the system as the mail server.

▼ How to Verify Mailhost Entries in DNS

The DNS name service does not support aliases for individuals. The DNS name service supports aliases for hosts or domains that use Mail Exchanger (MX) records and `CNAME` records. You can specify host names, domain names, or both in the DNS database. For more information, see “How `sendmail` Works With Name Services” on page 45, or *Working With Oracle Solaris 11.4 Directory and Naming Services: DNS and NIS*.

1. **Become an administrator.**

   For more information, see “Using Your Assigned Administrative Rights” in *Securing Users and Processes in Oracle Solaris 11.4*.

2. **Check for the `mailhost` and `mailhost.`*domain* entry in the `/etc/hosts` file.**

3. **Use `nslookup` to ensure that an entry exists for `mailhost` and `mailhost.`*domain* in the DNS database.**

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

◆ ◆ ◆ CHAPTER 5

Administering Mail Services

This chapter provides information about how to administer the files required for a functioning mail service.

The chapter covers the following topics:

- "Changing the `sendmail` Configuration" on page 53
- "Administering Mail Alias Files" on page 64
- "Administering the Mail Queue Directories" on page 69
- "Administering `.forward` Files" on page 72

Changing the `sendmail` Configuration

You can still use older versions of `sendmail.cf` files but the best practice is to use the new version. The `sendmail.cf` file consists of settings and rewriting rules that get upgraded with every new version. Rebuilding the `sendmail.cf` file results in the tweaks being applied on top of the new settings and rules. Hence, it is recommended to use the new version of the `sendmail.cf` file.

For more details, see the following sources of information:

- The `/etc/mail/cf/README` file provides a complete description of the configuration process
- For information about the configuration file, see "Versions of the Configuration File" on page 35 and "`sendmail` Configuration File" on page 34
- For additional information, see "Additional and Revised `m4` Configuration Macros in Version 8.12 of `sendmail`" on page 101

Tasks for changing the configuration are as follows:

- "How to Build a New `sendmail.cf` File" on page 54
- "Setting Up a Virtual Host" on page 55

- “Automatically Rebuilding Configuration Files” on page 56
- “Using `sendmail` in Open Mode” on page 56
- “How to Set SMTP to Use TLS” on page 57
- “How to Manage Mail Delivery by Using an Alternate Configuration of `sendmail.cf`” on page 63

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

1. **Become an administrator.**

   For more information, see “Using Your Assigned Administrative Rights” in *Securing Users and Processes in Oracle Solaris 11.4*.

2. **Stop the `sendmail` service.**

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

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

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

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

4. **Edit the new configuration files.**

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

   ```
   # cat hostname.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 hostname.cf
   ```

6. **Test the new configuration file.**

   ```
   # /usr/lib/sendmail -C hostname.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. For more information, see "Testing a New Mail Configuration" on page 77.

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

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

8. **Restart the `sendmail` service.**

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

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

Setting Up a Virtual Host

If you need to assign more than one IP address to a host, use `sendmail` to set up a virtual host.

Follow the instructions in Virtual Hosting except that in the "Sendmail Configuration" section, do **not** perform Step 3.b.:

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

Instead, use the following commands to set up a virtual host for the Oracle Solaris OS:

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

Use the name of the `.cf` file for the value of *mailserver*.

For additional information, see the description of `virtusertable` in `/etc/mail/cf/README`, and see the `sendmail`(5) man page to learn how to modify the `sendmail.cf` configuration file.

Automatically Rebuilding Configuration Files

If you have built your own copy of `sendmail.cf` or `submit.cf`, the configuration file is not rebuilt during the upgrade process. You can configure the `sendmail` service properties so that the `sendmail.cf` file is automatically rebuilt for you. You must have administrator privileges.

- To set the `sendmail` properties to automatically rebuild the `sendmail.cf` file, issue the following commands:

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

- Refresh and restart the `sendmail` service.

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

  The `svcadm refresh` command pushes the changes into the running snapshot. The `svcadm restart` command restarts the `sendmail` service using the new options.

To configure the `sendmail-client` service so that the `submit.cf` configuration file is rebuilt automatically from a `custom.mc` file, issue the following commands:

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

Using `sendmail` in Open Mode

The `sendmail` service runs in local-only mode by default so that only mail from the local host is accepted. Messages from any other systems are rejected. Open mode accepts incoming mail from all remote systems.

Caution - Running `sendmail` in local-only mode is much more secure than running in open mode. Make sure that you are aware of the potential security risks.

In order to set up `sendmail` in open mode, you must have administrator privileges.

To change the `sendmail` properties to allow incoming mail from remote systems, issue the following commands:

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

Refresh and restart the `sendmail` service to make the change take effect.

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

Setting SMTP to use TLS

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

▼ How to Set SMTP to Use TLS

This 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 87.

Before You Begin You need to provide the following information to create certificates:

- Country Name
- State or Province Name
- Locality Name
- Organization Name
- Organizational Unit Name
- Common Name, which is the machine's fully qualified host name. For more information, see the `check-hostname`(8) man page.
- Email Address

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

2. **Stop the `sendmail` service.**

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

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

    a. **Run the following commands:**

    ```
    # 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. **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.**

    The following example command 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`	Creates and processes certificate requests.
`-new`	Generates a new certificate request.
`-x509`	Creates a self-signed certificate.
`-keyout private/cakey.pem`	Assigns `private/cakey.pem` as the file name for the newly created private key.
`-out cacert.pem`	Assigns `cacert.pem` as the output file.
`-days 365`	Certifies the certificate for `365` days. The default value is `30`.
`-config openssl.cnf`	Specifies `openssl.cnf` as the configuration file.

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.**

   For example:

```
# 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.

In this example, the certificate and private key are in the file `newreq.pem`.

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

For example:

```
# 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` applications	Signs certificate requests in a variety of forms and generates certificate revocation lists (CRLs)

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 88.

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

For more information, see "Changing the `sendmail` Configuration" on page 53.

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. **Deny read permission to group and others for `MYkey.pem` for more security.**

```
# 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 the `sendmail` service.**

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

Example 1 `Received:` Mail Header

The following example shows a sample `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 [192.0.2.0])
        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 90.

For more information, see the following OpenSSL man pages:

- openssl(1)
- req(1)
- x509(1)
- ca(1)

Managing SMTP Mail Delivery

To facilitate the transport of inbound mail and outbound mail, the 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, use an alternate configuration of `sendmail.cf`.

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

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

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

2. **Stop the `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-host-domain.mc
   ```

4. **Edit the new configuration file.**

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

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

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

   ```
   # make submit-host-domain.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-host-domain.cf /etc/mail/submit.cf
   ```

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

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

   For more information, see "`submit.cf` Configuration File in Version 8.12 of `sendmail`" on page 95.

Administering Mail Alias Files

Mail aliases must be unique within the domain. The following table points to the procedures for administering mail alias files. For more information, see "Mail Alias Files" on page 35.

Task	Description	For Instructions
Set up an NIS `mail.aliases` map	Creates a `mail.aliases` map if you use the NIS name service.	"How to Set Up an NIS `mail.aliases` Map" on page 64
Set up a local mail alias file	Creates an `/etc/mail/aliases` file if you are not using any name service.	"How to Set Up a Local Mail Alias File" on page 65
Create a keyed map file	Facilitates aliasing with a keyed map file.	"How to Create a File to Redirect Addresses" on page 67
Set up the `postmaster` alias	Creates a `postmaster` alias to which users can address questions and report problems with the mail system.	"Managing the `postmaster` Alias" on page 67

▼ How to Set Up 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 "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

3. **Add the following entries to the `/etc/mail/aliases` file:**

   - 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*`.example.com`)

- A `Postmaster: root` entry

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

- An alias for `root` using the mail address of the person who is designated as the postmaster

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

user@host`.example.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. For more information, see Chapter 6, "Setting Up and Configuring Network Information Service" in *Working With Oracle Solaris 11.4 Directory and Naming Services: DNS and NIS*.**

5. **Change to the `/var/yp` directory and issue the `make` command to reflect changes on the NIS slave systems.**

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

The changes in the `/etc/hosts` and `/etc/mail/aliases` files are propagated to NIS slave systems.

How to Set Up 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 "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

3. **Add the following entries to the `/etc/mail/aliases` file:**
   - 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

   - A `Postmaster: root` entry

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

   - An alias for `root` using the mail address of the person who is designated as the postmaster

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

user@host`.example.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 the command generates either a single file, `/etc/mail/aliases.db`, or a pair of files, `/etc/mail/aliases.dir` and `/etc/mail/aliases.pag`.

5. **Perform one of the following steps to the file or files that are generated.**
   - **Copy the `/etc/mail/aliases`, the `/etc/mail/aliases.dir`, and the`/etc/mail/aliases.pag` files to the `/etc/mail` directory of each of the other systems.**
   - **Copy the `/etc/mail/aliases` and the `/etc/mail/aliases.db` files to the `/etc/mail` directory of each of the other systems.**

     You can copy the file or files generated by `rsync` command. For more information, see the `rsync(1)` man page. Alternatively, 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, you must update all the `/etc/mail/aliases` files each time you add or remove a mail client.

▼ How to Create a File to Redirect Addresses

You create a keyed map file to redirect addresses from one user and domain to another.

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

2. **Create a map input file.**

   Create a map input file with the following three entry types:

   - An entry to redirect mail to a new alias

     old-name@*newdomain*`.example.com` *new-name*@*newdomain*`.example.com`
   - An entry indicating a message to send when an incorrect alias is used

     old-name@*olddomain*`.example.com    error:nouser No such user here`
   - An entry redirecting all incoming mail from the old domain to the new domain

     `@`*olddomain*`.example.com          %1@`*newdomain*`.example.com`

   where *olddomain*`.example.com` is the domain that was previously assigned and *newdomain*`.example.com` is the newly assigned domain.

3. **Create the database file.**

   ```
   # /usr/sbin/makemap map-type new-map < new-map
   ```

map-type	Select a database type, such as `dbm`, `btree`, or `hash`.
new-map	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. This section provides procedures that describe how to create an NIS alias for `postmaster`, or create the alias in each local `/etc/mail/aliases` file.

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

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

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. **In each system's `/etc/mail/aliases` file, change `root` to the mail address of the person who is designated as the postmaster.**

   ```
   Postmaster: postmaster-mail-address
   ```

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.

▼ How to Create a Separate Mailbox for `postmaster`

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

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

   For more information, see "Setting Up and Managing User Accounts (Task Map)" in *Managing User Accounts and User Environments in Oracle Solaris 11.4*.

3. **Enable the `mail` program to read and write to the mailbox name after mail has been delivered.**

   ```
   # mail -f postmaster-mail-address
   ```

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

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

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.example.com
```

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

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

```
# cat /etc/mail/aliases
..
new-alias-name: /usr/path
```

new-alias-name	Create a name for a new alias
/usr/*path*	Use the path to the local mailbox

4. **Rebuild the alias database.**

```
# newaliases
```

Administering the Mail Queue Directories

This section describes some helpful tasks for queue administration. For information about the client-only queue, see "`submit.cf` Configuration File in Version 8.12 of `sendmail`" on page 95. For other related information, you can see "Additional Queue Features in Version 8.12 of `sendmail`" on page 107.

Tasks for administering the mail queue are as follows:

- "Displaying the Contents of the Mail Queue, `/var/spool/mqueue`" on page 70

- “Forcing Mail Queue Processing in the `/var/spool/mqueue` Mail Queue” on page 70
- “Running a Subset of the `/var/spool/mqueue` Mail Queue” on page 71
- “How to Move the `/var/spool/mqueue` Mail Queue” on page 71
- “How to Run the `/var/spool/omqueue` Old Mail Queue” on page 72

Displaying the Contents of the Mail Queue, `/var/spool/mqueue`

You can display the messages that are in the queue by using the following command:

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

This command provides the following information:

- Queue IDs
- Size of the message
- Date that the message entered the queue
- Message status
- Sender and the recipients

Additionally, this command checks for the authorization attribute, `solaris.admin.mail.mailq`. If the attribute is found, the equivalent of specifying the `-bp` flag with `sendmail` is executed. If the attribute is not found, 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, see the `prof_attr`(5) and `mailq`(1) man pages.

Forcing Mail Queue Processing in the `/var/spool/mqueue` Mail Queue

You can process messages to a system that was previously unable to receive messages.

You will have to become an administrator. For more information, see “Using Your Assigned Administrative Rights” in *Securing Users and Processes in Oracle Solaris 11.4*.

After becoming an administrator, use the following command to force queue processing and display the progress of the jobs as the queue is cleared.

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

Running a Subset of the `/var/spool/mqueue` Mail Queue

You use this command to force a substring of an address, such as a host name, to be processed. You also use this procedure to force a particular message from the queue.

You will have to become an administrator. For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

You can then use the following command to 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

Alternatively, you can use the following command to run a subset of the mail queue with `-qI`*nnnnn*.

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

nnnnn	Use a queue ID

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

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

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

2. **Stop the `sendmail` service so that it is no longer processing the queue directory.**

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

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 the `sendmail` service.**

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

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

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

2. **Run the old mail queue.**

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

`-oQ`	Specifies an alternate queue directory
`-q`	Runs every job in the queue
`-v`	Displays verbose output on the screen

3. **Remove the empty directory.**

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

Administering `.forward` Files

The following table describes the procedures for administering `.forward` files. For more information, see "`.forward` Files" on page 38.

Task	Description	For Instructions
Disable `.forward` files	Prevents automated forwarding.	"How to Disable `.forward` Files" on page 73
Change the `.forward` file search path	Moves all `.forward` files into a common directory.	"How to Change the `.forward` File Search Path" on page 74
Create and populate `/etc/shells`	Enables users to use the `.forward` file to forward mail to a program or to a file.	"How to Create and Populate the `/etc/shells` File" on page 74

▼ How to Disable `.forward` Files

This procedure prevents automated forwarding by disabling the `.forward` file for a particular host.

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

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 filename.m4
   ```

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.**

   For more information, see "How to Build a New `sendmail.cf` File" on page 54.

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

▼ How to Change the `.forward` File Search Path

If you want to put all `.forward` files in a common directory, use the following instructions.

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

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 filename.m4
   ```

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.**

   For more information, see "How to Build a New `sendmail.cf` File" on page 54.

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

Assigning Shell Access for `.forward` Permission

The `/etc/shells` 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 uses a script that can be downloaded, which is easier to use.

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

1. **Download the script.**

2. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

3. **Generate a list of shells.**

   ```
   # ./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. **Remove any shells from the list of shells in `/tmp/shells` that you are not including.**

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

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

Switching Mail Transfer Agents

Oracle Solaris enables you to switch between mail transfer agents (MTAs) such as `sendmail` and `postfix`. For more information about using `postfix`, please refer to the `postfix` documentation at `http://www.postfix.org/documentation.html`.

▼ How to Install and Run the `postfix` Service

1. **Disable the `sendmail` service.**

   ```
   # svcadm disable sendmail
   ```

2. **Disable the `sendmail` client.**

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

3. **Change the `sendmail` mediator to the `postfix` implementation.**

   ```
   # pkg set-mediator -I postfix sendmail
   ```

4. **Configure your domain name.**

   ```
   # postconf mydomain=YOUR.DOMAIN
   ```

Note - If your host name is fully qualified, such as `host.example.com`, then you can skip the `postconf` step.

Note - You can determine your domain name, *YOUR.DOMAIN* in the following ways based on the decreasing order of preference:

- Type the following command:

  ```
  svcprop -cp config/domain dns/client
  ```
- Type the following command:

  ```
  svcprop -cp config/search dns/client | nawk '{print $1}'
  ```

---

5. **Enable the `postfix` client.**

   ```
   # svcadm enable postfix
   ```

▼ How to Revert to the `sendmail` Service

1. **Disable the `postfix` service.**

   ```
   # svcadm disable postfix
   ```

2. **Change the `sendmail` mediator back to the `sendmail` implementation.**

   ```
   # pkg set-mediator -I sendmail sendmail
   ```

3. **Enable the `sendmail` client.**

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

4. **Enable the `sendmail` service.**

   ```
   # svcadm enable sendmail
   ```

◆ ◆ ◆ CHAPTER 6

Troubleshooting Mail Services

This chapter provides information about troubleshooting, and understanding and resolving error messages related to mail services:

- "Troubleshooting Procedures and Tips for Mail Services" on page 77
- "Error Messages" on page 80

Troubleshooting Procedures and Tips for Mail Services

Follow these procedures and tips to troubleshoot the mail service:

- "Testing a New Mail Configuration" on page 77
- "Verifying Mail Aliases" on page 78
- "How to Test the `sendmail` Rule Sets" on page 78
- "Verifying Connections to Other Systems" on page 79
- "Other Sources for Mail Diagnostic Information" on page 80

Testing a New Mail Configuration

You can perform various actions test the changes that you make to your configuration file.

First, restart the `sendmail` service on any system that has a revised configuration file.

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

Try the following actions to test the new configuration:

- 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.

- Send mail to yourself or other people on the local system by addressing the message to a regular user name.
- If you are connected to a network, send mail from the following systems to other users on different systems:
    - From the main system to a client system
    - From a client system to the main system
    - From a client system to another client system
- 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.
- Send messages from different systems to `postmaster` and ensure that the messages are delivered to your postmaster's mailbox.

Verifying Mail Aliases

You need to use the `mconnect` program to verify mail aliases. In the following example, the `mconnect` program opens a connection to a mail server on a local host and enables you to test that connection. The program runs interactively, so you can issue various diagnostic commands. The entry `expn sandy` provides the expanded address, `sandy@phoenix.example.com`. This entry verifies that messages using the alias `sandy` will also be delivered. For more information, see the `mconnect(1)` man page.

```
% mconnect
connecting to host localhost (127.0.0.1), port 25
connection open
220 your.domain.example.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.example.com closing connection
%
```

▼ How to Test the `sendmail` Rule Sets

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

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

2. **Test a mail address.**

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

   ```
   > 3,0 mail-address
   ```

3. **End the session.**

   Press Control-D.

Example 2 Address Test Mode Output

This example shows sample 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 . >
```

Verifying 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. For more information, see the `mconnect(1)` man page. The following example verifies that mail sent 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 the following conditions:

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

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 every night. 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 and outgoing messages.

Error Messages

The mail service uses the `syslogd` program to log error messages. These messages are logged for reference when resolving issues.

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 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` program 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 immediately
- `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

For more information, see the `syslog.conf`(5) man page.

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. The lowest level logs only unusual occurrences. The highest level logs every event that takes place. As a convention, log levels under 10 are considered to be useful. Log levels that are higher than 10 are usually used for debugging. The default level (9) has the right balance for most sites. For more information, see "Customizing System Message Logging" in *Troubleshooting System Administration Issues in Oracle Solaris 11.4*.

Resolving Error Messages

This section describes how you can resolve some common `sendmail`-related error messages. For more information, see Open Source Frequently Asked Questions (`https://web.archive.`

`org/web/20161027034616/http://www.sendmail.com:80/sm/open_source/support/support_faq)`.

Error: `451 timeout waiting for input during` *source*

Cause: When `sendmail` reads from a source that might time out, such as an SMTP connection, the program sets a timer to a 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. For more information, see the `sendmail`(8) man page.

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, check for hardware problems such as poor network cabling or connections.

Error: `550` *hostname*`... Host unknown`

Cause: The destination host machine, which is specified by the portion of the address after the `@` sign (at), 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. Also check whether you have typed the wrong domain. If the domain does not exist, contact the intended recipient and ask for a proper address.

Error: `550` *username*`... User unknown`

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

Solution: Check whether you have typed the wrong email address and try again. If this remedy does not work, contact the intended recipient and ask for a proper address.

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

Cause: 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 the 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 sending system's name. If these names are the same, the receiving system issues the error message and closes the connection. The name that is provided in the `HELO` command is the value of the `$j` macro.

Error: `config error: mail loops back to myself.`

Cause: This error message occurs when you set up an MX record. Host *host1* is made the mail exchanger for domain *domain1* but the host is not configured.

Another possibility is that both the sending system and the receiving system are identified as the same domain.

Solution: Search in Open Source Frequently Asked Questions (`https://web.archive.org/web/20161027034616/http://www.sendmail.com:80/sm/open_source/support/support_faq`).

Error: `user unknown`

Cause: The intended recipient on the same system cannot be located.

Solution: Some possible solutions:

- Check for a typographical error in the entered email address. The user might be aliased to a nonexistent email address in `/etc/mail/aliases` or in the user's `.mailrc` file.
- The user name might use uppercase characters. Preferably, email addresses should not be case sensitive.

◆ ◆ ◆ CHAPTER 7

Changes in Versions of `sendmail`

This chapter provides information about the changes in the different versions of `sendmail` and covers the following topics:

- "Changes in Version 8.15 of `sendmail`" on page 85
- "Changes in Version 8.14 of `sendmail`" on page 86
- "Changes in Version 8.13 of `sendmail`" on page 86
- "Changes in Version 8.12 of `sendmail`" on page 94

Changes in Version 8.15 of `sendmail`

A significant change to sendmail in version 8.15 is the inclusion of `ClientSSLOptions`, the SSL related options for the client side, and `ServerSSLOptions`, the SSL related options for the server side. This feature enables `sendmail` to narrow the SSL options used, whether acting as a client or a server. The feature is also useful in restricting older versions of SSL, resulting in increased security.

You can configure `ServerSSLOptions` in the system's `.mc` file. For example, to the `.mc` file, add the following line:

```
define(`confSERVER_SSL_OPTIONS', `SSL_OP_NO_SSLv2,SSL_OP_NO_SSLv3')dnl
```

This configuration results in the following output in the `/etc/mail/sendmail.cf` file.

```
O ServerSSLOptions=SSL_OP_NO_SSLv2,SSL_OP_NO_SSLv3
```

For more information, see the `sendmail(8)` man page. You can also see the `NOTES` section of `SSL_CTX_set_options(3openssl)` man page.

Changes in Version 8.14 of `sendmail`

Some significant changes to sendmail in version 8.14 are:

- The system can be configured to automatically rebuild the `sendmail.cf` and the `submit.mc` configuration files. The required steps are documented in "Automatically Rebuilding Configuration Files" on page 56.
- By default, the sendmail daemon runs in the new local daemon mode. The local-only mode accepts only 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. Use the `-bl` option to select the local-only mode, also known as the Become Local mode. For more information about this mode, see the `sendmail`(8) man page. For instructions about how to change back to the `-bd` or Become Daemon mode, see "Using `sendmail` in Open Mode" on page 56.
- 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 when used 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`(8) man page for more information about these options.

Changes in Version 8.13 of `sendmail`

Version 8.13 of `sendmail` provides many new features. For more information about version 8.13, see the following sections:

- "`FallBackSmartHost` Option" on page 86
- "Additional Command-Line Options in Version 8.13 of `sendmail`" on page 92
- "Additional and Revised Configuration File Options in Version 8.13 of `sendmail`" on page 92
- "Additional and Revised `FEATURE()` Declarations in Version 8.13 of `sendmail`" on page 94

`FallBackSmartHost` Option

The `FallBackSmartHost` option means you no longer need to use the `main.cf` and `subsidiary.cf` files. 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.

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 messages:

- `220 Ready to start TLS` - Requires the client to start the TLS negotiation.
- `501 Syntax error (no parameters allowed)` - Indicates that the client incorrectly issued the `STARTTLS` command. `STARTTLS` is issued with no parameters.
- `454 TLS not available due to temporary reason` - Requires the client to apply rule set values to determine whether to accept or maintain the connection.

Note that to maintain the Internet's SMTP infrastructure, servers available to the public 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 client must 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. Because most SMTP connections are not secure, the server and client might maintain an unsecured 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 57. 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 configuration file options are used to run SMTP with TLS:

`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.
`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.

If you declare any of the preceding 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

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 macros are used by the `STARTTLS` command:

`${cert_issuer}`	Holds the distinguished name (DN) of the certification authority (CA), which is the certificate issuer.
`${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.

`${server_addr}`	Holds the address of the server with the current outgoing SMTP connection.
`${server_name}`	Holds the name of the server with the current outgoing SMTP connection.

Rule Sets for Running SMTP With TLS

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

`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.
`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.

Security Considerations Related to Running SMTP With TLS

Note the following 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`.
- Because mail might be routed through multiple servers, for complete SMTP security, the entire chain of SMTP connections must have TLS support.
- Consider the level of negotiated authentication and privacy between each pair of servers or a client and server pair. For more information, see Chapter 1, "Using Secure Shell" in *Managing Secure Shell Access in Oracle Solaris 11.4*.

Additional Command-Line Options in Version 8.13 of `sendmail`

The following additional command-line options are available in version 8.13 of `sendmail`. For other command-line options, see the `sendmail(8)` man page.

`-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*.
`-Q`*reason*	Quarantines a normal queue item with this *reason* value. 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* value.

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

The added and revised configuration file options in version 8.13 are as follows:

`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.

`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.

If you declare any of the preceding 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

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

The `m4` macro uses the following syntax:

`FEATURE(`` ` ``*name*`', `` ` ``*argument*`')`

The following `FEATURE()` declarations in version 8.13 have been added and revised:

`conncontrol`	Works with the `access_db` rule set to check the number of incoming SMTP connections.
`greet_pause`	Adds the `greet_pause` rule set, which enables open proxy and SMTP slamming protection.
`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 the experimental support for MTAMark feature.
`ratecontrol`	Works with the `access_db` rule set to control connection rates for hosts.
`use_client_ptr`	If this `FEATURE()` is enabled, the rule set `check_relay` overrides its first argument with this argument, `$&{client_ptr}`.

For more information, see `/etc/mail/cf/README`.

Changes in Version 8.12 of `sendmail`

This section contains information about changes in version 8.12 of sendmail.

Support for TCP Wrappers in 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 `ftpd`.

Starting with version 8.12, `sendmail` enabled the use of TCP wrappers. This check does not bypass other security measures but rather validates the source of a network request before the request is granted. See the `hosts_access`(5) man page.

For information about ACLs, see "Using Access Control Lists to Protect UFS Files" in *Securing Files and Verifying File Integrity in Oracle Solaris 11.4*.

`submit.cf` Configuration File in Version 8.12 of `sendmail`

Starting with version 8.12, `sendmail` included 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 method provides better security.

Note the following information 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(8)` man page.
- `submit.cf` is used in the following operating modes:
  - `-bm`, the default operating mode
  - `-bs`, which uses standard input to run SMTP
  - `-bt`, the test mode that is used to resolve addresses
- When using `submit.cf`, `sendmail` does not run as an SMTP daemon.
- When using `submit.cf`, `sendmail` 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:

- `submit.cf` is provided automatically.
- `submit.cf` does not require any planning or preliminary procedures prior to Oracle Solaris installation.
- 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
```

Other distinguishing functions for `sendmail.cf` include the following:

- 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 in 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 Mail Queue Directories" on page 69.
- 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 provide misleading output. No workaround is available.
- Prior to `sendmail` version 8.12, not running `sendmail` in daemon mode would prevent the delivery only of inbound mail. Starting with `sendmail` version 8.12, not running the

`sendmail` daemon with the default configuration also prevents 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 63 for a way to resolve this problem.

Additional or Deprecated Command-Line Options in Version 8.12 of `sendmail`

The following list describes additional or deprecated command-line options for `sendmail`. For other command-line options, see the `sendmail`(8) man page.

`-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 in Version 8.12 of `sendmail`” on page 95.
`-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 in Version 8.12 of `sendmail`” on page 95.
`-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 canonification is done.
`-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. To process only jobs that do not have this substring of one of the recipients, add the `!` character.
`-q[!]R` *substring*	Processes only jobs that contain this substring of the queue ID. To process only jobs that do not have this substring of the queue ID, add the `!` character.

`-q[!]S` *substring*	Processes only jobs that contain this substring of the sender. To process only jobs that do not have this substring of the sender, add the `!` character.
`-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.
`-U`	As is noted in the Release Notes that are part of the `sendmail` distribution.

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

The following list describes additional macro-processed arguments for the `PidFile` and `ProcessTitlePrefix` options. For more information about these options, see the `sendmail`(8) man page.

`${daemon_addr}`	Provides daemon address (for example, 0.0.0.0)
`${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 in Version 8.12 of `sendmail`

The following additional macros are reserved for use by the `sendmail` program. The macro values are assigned internally. For more information, refer to the `sendmail`(8) man page.

`${addr_type}`	Identifies the current address as an envelope sender or a recipient address.
`${client_resolve}`	Holds the result of the resolve call for `${client_name}`: `OK`, `FAIL`, `FORGED`, or `TEMP`.
`${deliveryMode}`	Specifies the current delivery mode that `sendmail` is using instead of the value of the `DeliveryMode` option.

`${dsn_envid}`
`${dsn_notify}`
`${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_family_out}`
`${if_name_out}`
	Avoids the reuse of `${if_addr}`. Holds the following values respectively:
	■ The address of the interface for the outgoing connection
	■ The family of the interface for the outgoing connection
	■ The host name 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.
`${msg_size}`	Holds the value of the message size (`SIZE=parameter`) in an ESMTP dialogue before the message has been collected. Thereafter, the macro holds the message size as computed by `sendmail` and is used in `check_compat`.
`${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 for the Configuration File in Version 8.12 of `sendmail`

The following additional macros are used to build the `sendmail` configuration file:

`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 in 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.

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 in Version 8.12 of `sendmail`

The following is a list of additional and revised `m4` configuration macros for `sendmail`:

`FEATURE()`

For details, refer to “Changes to the `FEATURE()` Declaration in Version 8.12 of `sendmail`” on page 101.

`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()`
`VIRTUSER_DOMAIN_FILE()`

When these macros are used, include `$={VirtHost}` in `$=R`. `$=R` is the set of host names that are allowed to relay.

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 53.

Changes to the `FEATURE()` Declaration in Version 8.12 of `sendmail`

This section lists the specific changes to the `FEATURE()` declarations. The following changes are supported:

`compat_check`

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.

Argument: String.

`delay_checks`

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.

Arguments: `friend`, which enables a spam-friend test, or `hater`, which enables a spam-hater test.

`dnsbl`

Enables you to check the return values for DNS lookups. Note that this `FEATURE()` enables you to specify the behavior of temporary lookup failures. You can include it multiple times.

Argument: A maximum of two arguments:

- DNS server name
- Rejection message

`enhdnsbl`

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`.

Argument: Domain name.

`generics_entire_domain`

Can be used to apply `genericstable` to subdomains of `$=G`.

Argument: None.

`ldap_routing`

Implements LDAP address routing.

`local_lmtp`

Sets the delivery status notification (DSN) diagnostic-code type for the local mailer to the proper value of `SMTP`.

Argument: Path name of an LMTP-capable mailer. The default is `mail.local`, which is LMTP capable in this Oracle Solaris release.

`local_no_masquerade`

Enables you to avoid masquerading for the local mailer.

Argument: None.

`lookupdotdomain`

Can be used to look up the *.domain* in the access map.

Argument: None.

`nocanonify`

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.

Argument: `canonify_hosts` or nothing.

`no_default_msa`

Changes `sendmail`'s default setting from `m4`-generated configuration files to "listen" on several different ports, an implementation of RFC 2476.

Argument: None.

`nouucp`

Determines whether to allow the `!` token in the local part of an address.

Argument: `reject`, which does not allow the `!` token, or `nospecial`, which does allow the `!` token.

`nullclient`

Now provides the full rule sets of a normal configuration, allowing you to perform antispam checks.

Argument: None.

`preserve_local_plus_detail`

Enables you to preserve the `+detail` portion of the address when `sendmail` passes the address to the local delivery agent.

Argument: None.

`preserve_luser_host`

Enables you to preserve the name of the recipient host, if `LUSER_RELAY` is used.

Argument: None.

`queuegroup`

Enables you to select a queue group that is based on the full email address or on the domain of the recipient.

Argument: None.

`relay_mail_from`

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.

Argument: *domain* is an optional argument.

`virtuser_entire_domain`

Can now be used 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.

Argument: None.

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 53.

The following list describes `FEATURE()` declarations that are no longer supported and their replacements:

`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`
`sun_reverse_alias_nis`
`sun_reverse_alias_nisplus`

`` FEATURE(`genericstable'). ``

Changes to the `MAILER()` Declaration in 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:

- 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 `MAILER`s have been removed. You now have no required order for listing your `MAILER`s 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 24. If you need to build a new `sendmail.cf` file, refer to "Changing the `sendmail` Configuration" on page 53.

Additional Delivery Agent Flags in Version 8.12 of `sendmail`

```
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,
```

The following additional delivery agent flags are not set by default. 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 preceding example.

`%`	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`	Disables the ability of the mailer to send null characters (for example, `\0`).
`2`	Disables the use of ESMTP and requires that SMTP be used instead.
`6`	Enables mailers to strip headers to 7 bit.

Additional Equates for Delivery Agents in Version 8.12 of `sendmail`

Use the following syntax to append new equates or new arguments to the equates that already exist in the configuration file:

`M`*agent-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
```

The following example shows how to modify the definition of a value for `m4` configuration. This example places a limit of 1000 on the number of messages that are delivered per connection on an `smtp` mailer.

```
define(`SMTP_MAILER_MAXMSGS', `1000')
```

If you need to build a new `sendmail.cf` file, refer to "Changing the `sendmail` Configuration" on page 53.

Note - Typically, you modify the equate definitions in the `mailer` directory only when you fine-tune.

Additional equates for the `M` delivery-agent definition command are as follows:

`/=`	Specifies a directory to apply `chroot()` to before the mailer program is executed Argument: Path to a directory
`m=`	Limits the number of messages that are delivered per connection on an `smtp`, `local`, or `relay` mailer

	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
`W=`	Specifies the maximum time to wait for the return of the mailer after all data has been sent Argument: An increment of time

Additional Queue Features in Version 8.12 of `sendmail`

Additional queue features in version 8.12 are as follows:

- 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 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 Mail Queue Directories” on page 69.
- 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 Mail Queue Directories” on page 69.

Changes for LDAP in Version 8.12 of `sendmail`

The changes in the use of the Lightweight Directory Access Protocol (LDAP) in version 8.12 of sendmail are as follows:

- `LDAPROUTE_EQUIVALENT()` and `LDAPROUTE_EQUIVALENT_FILE()` enable 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 in `https://web.archive.org/web/20161028174456/http://sendmail.com/sm/open_source/docs/`. 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 shown in the following example:

  `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.

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 "*"

Additional LDAP map flags include:

`-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 in Version 8.12 of `sendmail`

As of version 8.12 of sendmail, the old `[TCP]` built-in mailer is no longer 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. This mailer might resemble the following example:

```
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. Possible values for the first mailer argument include:

`A=FILE`	Use for UNIX domain socket delivery
`A=TCP`	Use for TCP/IP connections
`A=IPC`	No longer available as a first mailer argument

Additional Rule Sets in Version 8.12 of `sendmail`

The following additional rule sets are available in version 8.12 of sendmail:

`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`).

Additional rule set features in version 8.12 of sendmail are as follows:

- 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 9 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 in Version 8.12 of `sendmail`

Note the following changes to files in version 8.12 of sendmail:

- 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 31. 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 “Additional Directories and Files Used for Mail Services” on page 41. 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 Used 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

A

`-Ac` option
  `sendmail` command, 97
agents
  mail, 24
aliases
  creating, 26, 27
  definition, 26
  `/etc/mail/aliases` file, 37, 37
  loops, 27
  NIS `aliases` map, 37
  verifying, 78
`aliases` file, 30
`aliases.db` file, 30, 66
`aliases.dir` file, 30, 66
`aliases.pag` file, 30, 66
alternative commands
  `sendmail` command, 16
`-Am` option
  `sendmail` command, 97
audio files
  mailbox space requirements and, 22
automatic mounting
  `/var/mail` directory, 22

B

backups
  mail servers and mailboxes, 22
`-bP` option
  `sendmail` command, 97

C

changing
  `/etc/shells` file, 74
  `.forward` file search path, 74
`check-hostname` script, 42
`check-permissions` script, 42
`check_eoh` rule set
  `sendmail` command, 109
`check_etrn` rule set
  `sendmail` command, 109
`check_expn` rule set
  `sendmail` command, 109
`check_vrfy` rule set
  `sendmail` command, 109
`clientmqueue` directory, 42
command-line options from version 8.12
  `sendmail` command, 95, 97, 98
`compat_check` `FEATURE()` declaration, 102
compilation flags
  `sendmail` command, 15
`confFORWARD_PATH` definition, 73, 74
configuration files
  `sendmail` command, 34
configuring
  mail gateways, 23
connections to other systems
  verifying, 79
creating
  `/etc/shells` file, 74
  keyed map file, 67
  `postmaster` alias, 67
  `postmaster` mailbox, 68

D

`delay_checks` `FEATURE()` declaration, 102
delivery agent flags from version 8.12
  `sendmail` command, 105
desktop-publishing files
  mailbox space requirements and, 22
disabling
  `.forward` files, 73
DNS name service
  `sendmail` program and, 52
`dnsbl` `FEATURE()` declaration, 102, 104
`domain` directory, 32
dot (.)
  in mailbox names, 25

E

`editmap` command, 42
`enhdnsbl` `FEATURE()` declaration, 102, 104
equates for delivery agents from version 8.12
  `sendmail` command, 106
error messages
  logging,, 81
  resolving, 81
  `sendmail` program, 81
`/etc/default/sendmail` file, 40
`/etc/hosts` file, 50, 51
`/etc/mail` directory
  contents of, 30
`/etc/mail/aliases` file, 25, 30, 36, 37
`/etc/mail/aliases.db` file, 30, 66
`/etc/mail/aliases.dir` file, 30, 66
`/etc/mail/aliases.pag` file, 30, 66
`/etc/mail/cf` directory
  contents of, 31
`/etc/mail/cf/cf/main.cf` file, 32
`/etc/mail/cf/cf/main.mc` file, 32
`/etc/mail/cf/cf/Makefile` file, 32
`/etc/mail/cf/cf/sendmail.cf` file, 32
`/etc/mail/cf/cf/sendmail.mc` file, 32
`/etc/mail/cf/cf/submit.cf` file, 32
`/etc/mail/cf/cf/submit.mc` file, 32
`/etc/mail/cf/cf/subsidiary.cf`, 32
`/etc/mail/cf/cf/subsidiary.mc` file, 32
`/etc/mail/cf/domain` directory, 32
`/etc/mail/cf/domain/generic.m4` file, 32
`/etc/mail/cf/domain/solaris-antispam.m4` file, 32
`/etc/mail/cf/domain/solaris-generic.m4` file, 32
`/etc/mail/cf/feature` directory, 33
`/etc/mail/cf/m4` directory, 33
`/etc/mail/cf/mailer` directory, 33
`/etc/mail/cf/ostype` directory, 33
`/etc/mail/cf/ostype/solaris2.m4` file, 33
`/etc/mail/cf/ostype/solaris2.ml.m4` file, 33
`/etc/mail/cf/ostype/solaris2.pre5.m4` file, 33
`/etc/mail/cf/ostype/solaris8.m4` file, 33, 33
`/etc/mail/cf/README` file, 32
`/etc/mail/cf/sh/check-hostname` script, 42
`/etc/mail/cf/sh/check-permissions` script, 42
`/etc/mail/helpfile` file, 31, 110
`/etc/mail/local-host-names` file, 31, 110
`/etc/mail/Mail.rc` file, 30
`/etc/mail/mailx.rc` file, 30
`/etc/mail/main.cf` file, 30
`/etc/mail/relay-domains` file, 30
`/etc/mail/sendmail.cf` file, 31
`/etc/mail/sendmail.ct` file, 110
`/etc/mail/sendmail.cw` file, 110
`/etc/mail/sendmail.hf` file, 110
`/etc/mail/sendmail.pid` file, 31
`/etc/mail/statistics` file, 31
`/etc/mail/submit.cf` file, 31, 95
`/etc/mail/subsidiary.cf` file, 31, 50
`/etc/mail/trusted-users` file, 31, 110
`/etc/shells` file, 74
`/etc/syslog.conf` file, 81
Ethernet
  testing mail configuration, 78
`etrn` script, 42

F

`.forward` files
  administering, 72
  changing search path, 74
  disabling, 73
  for users, 38
`.forward+`*detail* file, 40
`.forward.`*hostname* files, 39
`FEATURE()` declarations in version 8.12
  supported, 101
  unsupported, 104
`FEATURE()` declarations in version 8.13 of `sendmail`, 94
`feature` directory, 33

G

`-G` option
  `sendmail` command, 97
`gen-etc-shells` script, 74
`generic.m4` file, 32
`generics_entire_domain` `FEATURE()` declaration, 102
`genericstable` `FEATURE()` declaration, 104
`gethostbyname` command, 47

H

`helpfile` file, 31
  `sendmail` command, 110

I

`in.comsat` daemon, 42
IPv6 addresses and version 8.12
  `sendmail` command, 111

K

keyed map file, creating, 67

L

`-L` *tag* option
  `sendmail` command, 97
LDAP from version 8.12
  `sendmail` command and, 108
`ldap_routing` `FEATURE()` declaration, 102
local delivery agent, mail services, 24
local mail addresses, 26
local mail alias file, setting up, 65
`local-host-names` file, 31, 110
`LOCAL_DOMAIN()` `m4` configuration macro, 101
`local_lmtp` `FEATURE()` declaration, 102
`local_no_masquerade` `FEATURE()` declaration, 103
log levels
  `sendmail.cf` file, 35
`lookupdotdomain` `FEATURE()` declaration, 103
loops
  aliases, 27

M

`.mailrc` aliases, 36
`.mailrc` file, 27
`m4` directory, 33
macros from version 8.12
  defined macros (`sendmail`), 99
  `m4` configuration macros (`sendmail`), 101
  `MAX` macros (`sendmail`), 100
mail addresses
  % in, 26
  local, 26
  mail routing and, 45
mail alias files
  `.mailrc` aliases, 36
  administering, 64
  description, 35
  `/etc/mail/aliases` file, 36
mail clients
  definition, 22
`mail` command, 29
mail configuration
  local mail and a remote connection, 50

  local mail only, 49
  testing, 77
  typical
    components, 17
mail connections to other systems
  testing, 79
mail domains
  name service domains and, 46
  `sendmail.cf` file and, 46
mail exchanger (MX) records, 52
mail gateways
  configuring, 23
  definition, 22
  `sendmail.cf` file and, 23
  testing, 78
mail hosts
  description, 21
mail queue
  administering the queue directories, 69
  forcing mail queue processing, 70
  moving the mail queue, 71
  running a subset of, 71
  running the old mail queue, 72
mail routing
  mail addresses and, 45
mail servers, 22
  backups and mailboxes, 22
  description, 22
  mailboxes, 22
  mailboxes on, 22, 26
  space requirements for, 22
mail services, 64, 69, 72, 77
  changes to `sendmail` from version 8.12, 94
  changes to `sendmail` in version 8.13, 86
  changes to `sendmail` in version 8.14, 86
  changes to `sendmail` in version 8.15, 85
  hardware components
    mail client, 22
    mail gateway, 23
    mail host, 21
    mail server, 22
    required elements, 21
  planning your mail system, 49
  setting up, 51
  software components, 23
    local delivery agent, 24
    mail aliases, 26
    mail transfer agent, 24
    mail user agent, 24
    mailbox files, 25
    mailers, 24
mail transfer agents, 24
mail user agents, 24
`Mail.rc` file, 30
mailboxes
  files for, 25, 25, 42
  mail servers, 22
  mail servers and, 22
  space requirements for, 22
`MAILER()` declarations from version 8.12, 105
`mailer` directory, 33
MAILER-DAEMON messages, 80
mailers
  built-in (`sendmail`)
    `[TCP]` and `[IPC]`, 109
  definition, 24
  Simple Mail Transfer Protocol (SMTP) mailers, 24
`mailq` command, 29
`mailstats` command, 30
`mailx` command, 30
`mailx.rc` file, 30
`main-v7sun.mc` file, 110
`main.cf` file, 30, 32, 34
`main.mc` file, 32, 110
`Makefile` file, 32
`makemap` command, 42
`MASQUERADE_EXCEPTION()` `m4` configuration
macro, 101
`MAXBADCOMMANDS` macro
  `sendmail` command, 100
`MAXETRNCOMMANDS` macro
  `sendmail` command, 100
`MAXHELOCOMMANDS` macro
  `sendmail` command, 100
`MAXNOOPCOMMANDS` macro

    `sendmail` command, 100
`MAXVRFYCOMMANDS` macro
    `sendmail` command, 100
`mconnect` command, 30, 79
`mqueue` directory, 42
MX (mail exchanger) records, 52

N
name service domains
    mail domains and, 46
`newaliases` link, 42
NIS `aliases` map, 37
NIS `mail.aliases` map, setting up, 64
`no_default_msa` `FEATURE()` declaration, 103
`nocanonify` `FEATURE()` declaration, 103
`nouucp` `FEATURE()` declaration, 103
`nullclient` `FEATURE()` declaration, 103

O
`openssl` command and `sendmail`, 58
options in `sendmail` command
    command-line options from version 8.12, 95, 97, 98
    command-line options in version 8.13, 92
    configuration file options in version 8.13, 92
    `PidFile` option, 98
    `ProcessTitlePrefix` option, 98
`ostype` directory, 33
`owner-` prefix and mailbox names, 26
`owner-` prefix, mail alias with, 26
`owner-owner` and mailbox names, 26

P
percent sign (%) in mailbox names, 26
`PidFile` option
    `sendmail` command, 98
`postmaster` alias, creating, 67
`postmaster` mailbox
    creating, 68
    description, 25
    testing, 78
`praliases` command, 30
`preserve_local_plus_detail` `FEATURE()` declaration, 103
`preserve_luser_host` `FEATURE()` declaration, 103
`ProcessTitlePrefix` option
    `sendmail` command, 98

Q
`-q[!]I`*substring* option
    `sendmail` command, 97
`-q[!]R`*substring* option
    `sendmail` command, 97
`-q[!]S`*substring* option
    `sendmail` command, 98
`-qf` option
    `sendmail` command, 98
`-qG`*name* option
    `sendmail` command, 98
`-qp`*time* option
    `sendmail` command, 98
queue features from version 8.12
    `sendmail` command, 107
`queuegroup` `FEATURE()` declaration, 104

R
`-request` suffix and mailbox names, 25
`rbl` `FEATURE()` declaration, 104
`relay-domains` file, 30
`relay_mail_from` `FEATURE()` declaration, 104
`remote_mode` `FEATURE()` declaration, 104
`rmail` command, 30
rule sets
    testing, 78
    version 8.12 of `sendmail`, 109
running SMTP with TLS
    configuration file options for, 88
    description, 87
    macros for, 90

  rule sets for, 91
  security considerations related to, 91
  task information, 57

S

`sendmail` command
  alternative commands, 16
  as transfer agent, 13
  changes from version 8.12, 94
  changes in version 8.13, 86
  changes in version 8.14, 86
  changes in version 8.15, 85
  changes to file name or file location from version 8.12, 110
  command-line options from version 8.12, 95, 97, 98
  command-line options in version 8.13, 92
  compilation flags, 15
  configuration file options in version 8.13, 92
  delivery agent flags from version 8.12, 105
  equates for delivery agents from version 8.12, 106
  error messages, 81
  `/etc/mail/helpfile` file, 110, 110
  `/etc/mail/local-host-names` file, 110
  `/etc/mail/sendmail.ct` file, 110
  `/etc/mail/sendmail.cw` file, 110
  `/etc/mail/submit.cf`, 95
  `/etc/mail/trusted-users` file, 110
  `FEATURE()` declarations
    changes from version 8.12, 101
  `FEATURE()` declarations from version 8.12
    supported, 101
    unsupported, 104
  `FEATURE()` declarations in version 8.13, 94
  features, 13
  `.forward` files, 38
  `helpfile` file, 110, 110
  interactions of NIS and DNS, 47
  IPv6 addresses and version 8.12, 111
  LDAP from version 8.12, 108
  `local-host-names` file, 110
  macros
    defined macros from version 8.12, 99
    `m4` configuration macros from version 8.12, 101
    `MAX` macros from version 8.12, 100
  `MAILER()` declarations from version 8.12, 105
  mailers, built-in
    `[TCP]` and `[IPC]`, 109
  `main-v7sun.mc` file, 110
  `main.mc` file, 110
  name services and, 46
  NIS `aliases` map, 37
  Oracle Solaris version of, 16
  queue features from version 8.12, 107
  rule sets from version 8.12, 109
  `sendmail.ct` file, 110
  `sendmail.cw` file, 110
  `submit.cf` file, 95
  `subsidiary-v7sun.mc` file, 110
  `subsidiary.mc` file, 110
  TCP wrappers and, 95
  `trusted-users` file, 110
  working with programs, 42
`sendmail.cf` file, 31, 32
  alternate configuration for, 63
  building the configuration file, 53
  description, 34
  log levels, 35
  mail domains and, 46
  mail gateways and, 23
  mail hosts and, 34
  mail servers and, 34
  mailers, description, 24
  vendor setting, 35
  version level, 35
`sendmail.ct` file, 110
`sendmail.cw` file, 110
`sendmail.hf` file, 110
`sendmail.mc` file, 32
`sendmail.pid` file, 31, 41
`sendmail.st` file *See* `statistics` file
setting SMTP to use TLS, 57
setting up

  local mail alias files, 65
  NIS `mail.aliases` map, 64
  virtual host, 55
`SMART_HOST()` `m4` configuration macro, 101
SMTP (Simple Mail Transfer Protocol)
  mailers, 24
  `sendmail.cf` file, 96
SMTP and TLS
  configuration file options for, 88
  description, 87
  macros for, 90
  rule sets for, 91
  security considerations related to, 91
  task information, 57
software components
  troubleshooting, 78
`solaris-antispam.m4` file, 32
`solaris-generic.m4` file, 32, 73
`solaris11.m4` file, 33
`solaris2.m4` file, 33
`solaris2.ml.m4` file, 33
`solaris2.pre5.m4` file, 33
`solaris8.m4` file, 33
`statistics` file, 31
`submit.cf` file, 31, 32, 95
`submit.mc` file, 32
`subsidiary-v7sun.mc` file, 110
`subsidiary.cf` file, 31, 32, 50
`subsidiary.mc` file, 32, 110
`sun_reverse_alias_files` `FEATURE()` declaration, 104
`sun_reverse_alias_nis` `FEATURE()` declaration, 104
`sun_reverse_alias_nisplus` `FEATURE()` declaration, 104
`syslog.conf` file, 81
`syslogd` command, 42
`/system/volatile/sendmail.pid` file, 41

T
TCP wrappers
  `sendmail` command and, 95
testing
  mail aliases, 78
  mail configuration, 77
  mail connections to other systems, 79
  rule sets, 78
TLS and SMTP
  configuration file options for, 88
  description, 87
  macros for, 90
  rule sets for, 91
  security considerations related to, 91
  task information, 57
Transport Layer Security (TLS) and SMTP
  configuration file options for, 88
  description, 87
  macros for, 90
  rule sets for, 91
  security considerations related to, 91
  task information, 57
troubleshooting
  `.forward` file issues, 38
  mail aliases, 78
  mail connections to other systems, 79
  mail services, 77
  MAILER-DAEMON messages and, 80, 80
  rule sets, 78
  tips and, 77
  undelivered mail, 78
`trusted-users` file, 31, 110

U
`-U` option
  `sendmail` command, 98
undelivered messages
  troubleshooting, 78
underscore (_) in mailbox names, 25
user names, mailbox names and, 25
`/usr/bin` directory
  contents of, 29
`/usr/bin/mail` command, 29
`/usr/bin/mailq` command, 29
`/usr/bin/mailstats` command, 30

`/usr/bin/mailx` command, 30
`/usr/bin/mconnect` command, 30, 79
`/usr/bin/praliases` command, 30
`/usr/bin/rmail` command, 30
`/usr/lib` directory
  contents of, 33
`/usr/sbin/editmap` command, 42
`/usr/sbin/etrn` script, 42
`/usr/sbin/in.comsat` daemon, 42
`/usr/sbin/makemap` command, 42
`/usr/sbin/newaliases` link, 42
`/usr/sbin/syslogd` command, 42
command, 30
directories
  for mail services, 29

V
`vacation` command, 30
`vacation` message utility, 48
`/var/mail` directory, 49, 51
`/var/mail` file, 25
`/var/spool/clientmqueue` directory, 42
`/var/spool/mqueue` directory, 42
vendor setting
  specifying in `sendmail.cf` file, 35
version level
  specifying in `sendmail.cf` file, 35
virtual hosts, setting up, 55
`VIRTUSER_DOMAIN()` `m4` configuration macro, 101
`VIRTUSER_DOMAIN_FILE()` `m4` configuration macro, 101
`virtuser_entire_domain` `FEATURE()` declaration, 104