74 Mail Services

Table of Contents

Mail-Grundlagen

- **MTA - M**ail **T**ransfer **A**gent

  Programme unter Unix/Linux: Postfix, Sendmail, qmail, exim, smail

- **MDA - M**ail **D**elivery **A**gent oder **LDA** - **L**ocal **D**elivery **A**gent

  Programme unter Unix/Linux: `mail`, `procmail`, `local` (Postfix), `qmail-local`

- **MUA - M**ail **U**ser **A**gent
  - MUAs unter Unix/Linux: `mail`, `pine`, `mutt`, `kmail` (kde), `balsa` (gnome) `evolution` (gnome)

    `mail:`

    `mail` ist das einfachste mail-Programm unter Linux um Mails zu senden oder zu bekommen. Schon rein für Testzwecken ist es gut dieses Programm ein bisschen zu kennen.

    - **Mail senden**:

```
mail pierre@localhost
Subject: einfacher test
Das ist mein erstes Mail mit mail
.
EOT
```

    - **Mails lesen:**

```
mail
Mail version 8.1 6/6/93.  Type ? for help.
"/var/mail/pierre": 1 message 1 new
>N  1 pierre@globeall.de    Fri Mar 29 21:00  13/468
"einfacher test"
& 1                (Liest das Mail mit der Zahl 1 - erstes Mail)
Message 1:
From pierre@globeall.de  Fri Mar 29 21:00:59 2002
Delivered-To: pierre@localhost.linux.local
To: pierre@localhost.linux.local
Subject: einfacher test
Date: Fri, 29 Mar 2002 21:00:58 +0100 (CET)
From: pierre@globeall.de (Pierre Burri)

Das ist mein erstes Mail mit mail

& d                (Löscht das aktuelle Mail)
& q                (Beendet mail)
```

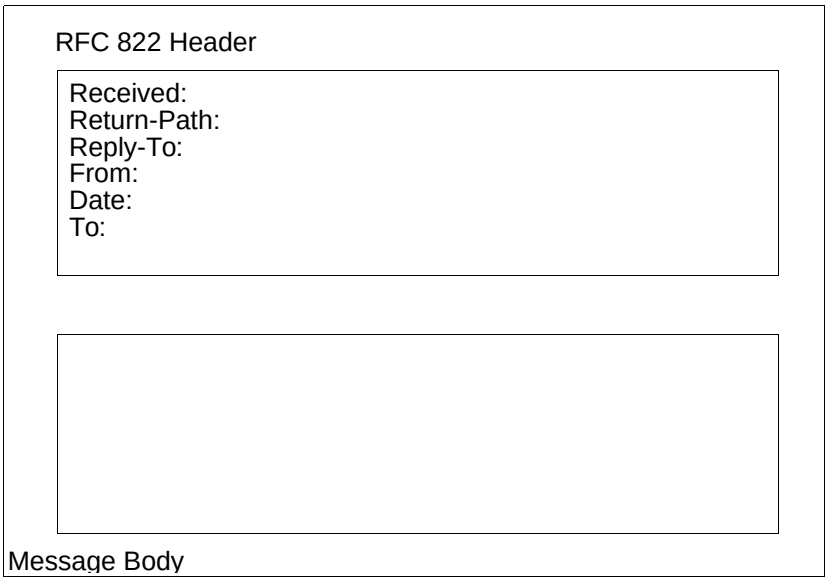
    Die Mails die gelesen worden sind werden automatisch in `$HOME/mbox` verschoben.

  - MUAs unter Windows: `Eudora`, `Outlook Express`, `MS Outlook`, `Netscape Composer`

- **UCE - U**ncolisited **C**ommercial **E**mail (Spam)

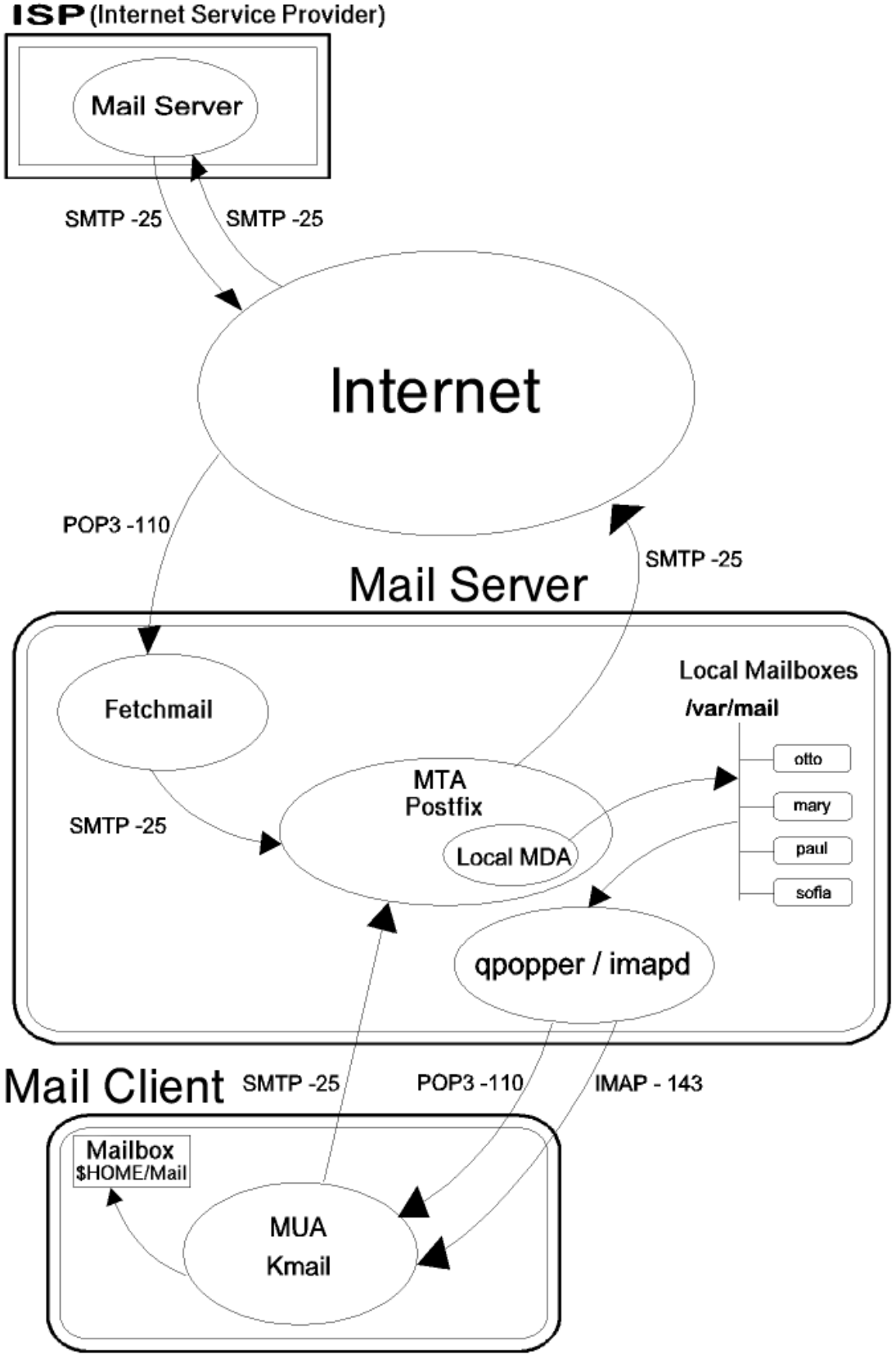
  UCE oder auch oft Spam genannt, steht für "unerwünschte kommerzielle Massen-E-Mail". UCEs sind meistens Werbe-Emails mit fragwürdigen Inhalten (viel Geld schnell verdienen, Porno-Angebote, illegale Informatinen usw.) die an so viel wie mögliche E-Mail-Adresse geschickt werden. UCEs kosten dem Sender kaum etwas, sind eine Belästigung und ein Missbrauch des Internets. Zum Glück ist es inzwischen möglich einen MTA gegen UCEs zu kongigurieren und zu schützen.

- **Aufbau einer E-Mail, RFC 822 Header**



- **Received:**
  Indentifiziert der Ursprüngliche Absender und alle Mail-Servern die das Mail weitergeleitet haben. Es kann dadurch mehrmals dieses Feld geben.
- **Return-Path:**
  Indentifiziert die Route die genommen wurde um das Mail zum letzten Mail-Server weiterzuleiten. Meistens steht hier die E-Mail-Adresse des Absenders.
- **Reply-To:**
  E-Mail-Adresse des Absenders oder
  die gewünschte E-Mail-Adresse um Antworten zu bekommen.
- **From:**
  Author des E-Mails bzw. die E-Mail-Adresse.
- **Date:**
  Datum und Zeit wann das E-Mail zum ersten Mail-Server gesendet wurde
- **To:**
  Empfänger des E-Mails. Diese Feld ist nur Informational. Einen SMTP-Server nimmt nur Empfänger an, für welche ein `RCPT` gegeben wurde.
- **CC: und BCC:**
  `Carbon Copy` (Kopie) und `Blind Carbon Copy` (Blindkopie). E-Mail-Adresse für einen Empfänger der eine Kopie des E-Mails bekommen soll. Bei `BCC` wird diesen Vorgang dem Hauptempfänger versteckt.

- **Mail-Dienste im Internet**



- ## Mail-Protokollen

  - **SMTP - Simple Mail Transfer Protocol (port 25)**

    SMTP-Befehlen: HELO, MAIL, RCPT, DATA, (SEND), (SOML), (SAML), RSET, VRFY, (EXPN), (HELP), NOOP, QUIT, (TURN).
    Die Befehlen in **()** sind bei Postfix nicht implementiert.

    Testen von SMTP mit telnet:

```
telnet servername 25
Trying 192.168.100.133...
Connected to 192.168.100.133.
Escape character is '^]'.
220 dozlinux.linux.local ESMTP Postfix

HELO laptop.linux.local
250 dozlinux.linux.local

MAIL From: me.linux.local
250 Ok

RCPT To: michel@dozlinux.linux.local
250 Ok

DATA
354 End data with <CR><LF>.<CR><LF>

Date: 01 Jan 2002 12:03:40
From: michel@laptop.linux.local
To: irmgard@dozlinux.linux.local
Subject: Hallo again!!

Hello Irmgard,

Bla bla bla, bis bald
.

250 Ok: queued as 0C5B32E9D

quit
221 Bye
```

  - **ESMTP -Extended Simple Mail Transfer Protocol (port 25)**

    ESMTP ist eine Erweiterung von SMTP und erlaubt mehr Befehle. Die meisten Mail-Server beherschen SMTP und ESMTP. ESMTP erlaubt eine Kommunikation über die gleiche Verbindung in beiden Richtungen. Das erlaubt z.B., die überprüfung des Mail-Servers der die Mail(s) über dein eigenen Mail-Server senden will. Eine ESMTP-Sitzung wird über den Befehl EHLO Rechnername gestartet. Spezielle Befehle des ESMTP-Protokoll sind z.B. ETRN Domänenamen (extended Turn), was das Holen von Mails von einem Mail-Server erlaubt und AUTH, was nach einer Authentifizieren erlaubt spezielle Befehle (z.B. Mail-Relay) auf dem Mail-Server auszuführen.

  - **POP3 - Post Office Protocoll Version 3 (Port 110)**

    POP3 ist das meist verbreite Protokoll heute um Mails von einem Server abzuholen. Es ist ein sehr einfaches Protokol.

Testen vom POP3 mit telnet:

Die fettschrifft sind die Eingegebene Befehle

telnet dozlinux.linux.local 110 (Server-Programm: ipop3d)

```
Trying 192.168.100.133
Connected to dozlinux.linux.local
Escape character is '^]'
+OK POP3 dozlinux.linux.local v2000.70 server ready
user Benutzername
+OK User name accepted, password please
pass Passwort
+OK Mailbox open, 2 messages
stat                          zeigt die Anzahl der Mails in der Mailbox und die
+OK  2  2019                  Grösse in Bytes
list                          gleich wie STAT, aber separat aufgelistet
+OK Mailbox scan listing follows
1 653
2 674
3 692
top 1 1                       zeigt der Header + die erste Zeile des ersten Mails
+OK Top of message follows
X-UIDL: +1b"!)&~"!&~)"!@:K!!
Return-Path: <root@globeall.de>
Delivered-To: pierre@dozlinux.linux.local
Received: from SUN.linux.local (sun.linux.local [192.168.100.44])
by dozlinux.linux.local (Postfix on SuSE Linux 7.3 (i386)) with ESMTP id 963B071E
for <pierre@dozlinux.linux.local>; Fri, 29 Mar 2002 10:51:19 +0100 (CET)
Received: by SUN.linux.local (Postfix, from userid 0)
 id 8D6081114; Fri, 29 Mar 2002 10:55:15 +0100 (CET)
To: pierre@dozlinux.linux.local
Subject: test pop3
Message-Id: <20020329095515.8D6081114@SUN.linux.local>
Date: Fri, 29 Mar 2002 10:55:15 +0100 (CET)
From: root@globeall.de (root)
Status:  OK

bla bla bla                                (das ist die erste Zeile)
.
retr 1                                     zeigt das ganze Mail Nr. 1
+OK 653 octets
(wieder das gleiche wie vorher aber mit dem ganzen Mail)

dele 1
+OK Message deleted           löscht das Mail Nr. 1
quit                          beendet die Verbindung zum Server
+OK Sayonara
Connection closed by foreign host.
```

- **IMAP - Interactive Mail Access Protocol (Port 143)**

  IMAP ist weniger bekannt als POP3 aber wird immer beliebter. Die letzte Version des Protokolls ist die Version 4 Revision 1, auch bekannt als IMAP4rev1.
  Der Hauptunterschied zu POP3 ist, dass die Mails auf dem Server bleiben. Das ist einen grossen Vorteil, weil Die Mails von verschieden Orten gelesen und verwaltet werden können.

  - **Testen von IMAP mit telnet:**

    telnet dozlinux.linux.local 143 (Das Server-Programm ist imapd)

```
Trying 192.168.100.133...
Connected to 192.168.100.133.
Escape character is '^]'.
* OK [CAPABILITY IMAP4 IMAP4REV1 STARTTLS LOGIN-REFERRALS AUTH=LOGIN]
dozlinux.linux.local IMAP4rev1 2000.287 at Fri, 29 Mar 2002 12:26:12
+0100 (CET)
```

    Achtung: Jeder Befehl muss mit einem sogenannten "Tag" (Kennzeichne) anfangen: a01, a02, a03 usw.

    a01 capability zeigt die "Fähigkeiten" des Programms

```
* CAPABILITY IMAP4 IMAP4REV1 STARTTLS NAMESPACE IDLE MAILBOX-REFERRALS
SCAN SORT THREAD=REFERENCES THREAD=ORDEREDSUBJECT MULTIAPPEND LOGIN-
REFERRALS AUTH=LOGIN
a01 OK CAPABILITY completed
```

    a02 login pierre passwort

```
* CAPABILITY IMAP4 IMAP4REV1 STARTTLS NAMESPACE IDLE MAILBOX-REFERRALS
SCAN SORT THREAD=REFERENCES THREAD=ORDEREDSUBJECT MULTIAPPEND
a02 OK LOGIN completed
```

    a04 select inbox öffnet eine Mailbox

```
* 2 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 1017395681] UID validity status
* OK [UIDNEXT 4] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS (\* \Answered \Flagged \Deleted \Draft \Seen)]
Permanent flags
* OK [UNSEEN 1] first unseen message in /var/spool/mail/pierre
a04 OK [READ-WRITE] SELECT completed
```

    a03 noop no operation. imapd zeigt was sich in der Mailbox (/var/mail/Benutzername) befindet. Wenn mbox existiert, werden die Mails nach mbox verschoben.

```
* 4 EXISTS
* 1 RECENT
a03 OK NOOP completed
```

    a05 FETCH 1 RFC822 zeigt das erste Mail

```
* 1 FETCH (RFC822 {2678}
Return-Path: <marty.volker@urz.uni-heidelberg.de>
Delivered-To: michel@localhost.linux.local
Received: from localhost (localhost [127.0.0.1])
.............................
.............................
 FLAGS (\Recent \Seen))
a05 OK FETCH completed
```

    18 fetch 1 flags zeigt der Zustand des ersten Mails

```
* 1 FETCH (FLAGS (\Seen))
18 OK FETCH completed
```

    a06 store 1 +flags (\deleted) markiert das Mail zum Löschen (-flags=wegnehmen)

```
* 1 FETCH (FLAGS (\Seen \Deleted))
a06 OK STORE completed
```

    a07 expunge

```
* 1 EXPUNGE
* 5 EXISTS
* 0 RECENT
a07 OK Expunged 1 messages

a08 LOGOUT
* BYE dozlinux.linux.local IMAP4rev1 server terminating connection
a08 OK LOGOUT completed
Connection closed by foreign host.
```

- **LMTP - Local Mail Transport Protocol**

  Der Vorteil von LMTP im Gegensatz zu SMTP, ist das es mehrere Status-Meldungen zu einem Mail das auch mehere Empfänger hat, zurückgeben kann. Der Sender weiss dann, nach einer Mailingliste-Verschikung, welche Empfänger haben die Mail bekommen oder nicht. Diese Protokoll kann z.B. zwischen einem MTA und einen MDA benutzt werden.

  Die LMTP-Befehle sind gleich wie bei SMTP/ESMTP aber es wird LHLO statt HELO oder EHLO benutzt um eine Sitzung zu öffnen.

- **Installation of Postfix**
  - Install the package `postfix` from SuSE CD
  - run the command `newaliases`
  - edit the file `/etc/postfix/main.cf`

    add the network interfaces to serve under:

    ```
    inet_interfaces = 127.0.0.1 1:: 192.168.70.130
    ```
  - restart postfix : `rcpostfix restart`

- **Testing postfix locally**
  - use mail program to send a mail to a local user

    ```
    mail username
    subject: test1 of postfix

    test1
    ^D  or .
    ```
  - `su - username`
    `mail`
    Sent Mail should be there

- **Testing postfix remotely**
  - Make sure the DNS is configured properly with MX records for destination domain
    [*dest.domain*] IN MX *order* *mail.server.domain*.
    order = order of connection attempts to servers when multiple
  - `mail username@remote.host.domain` (FQDN)
    subject.....
  - on the remote host:

    ```
    su - username
    mail
    ```
    Sent Mail should be there
- **To resend stuck mail from the mail queue:**
  `postfix flush`
  `mailq` (to check again if they are gone)
- **Postfix: Einen von vielen Mail-Servern**

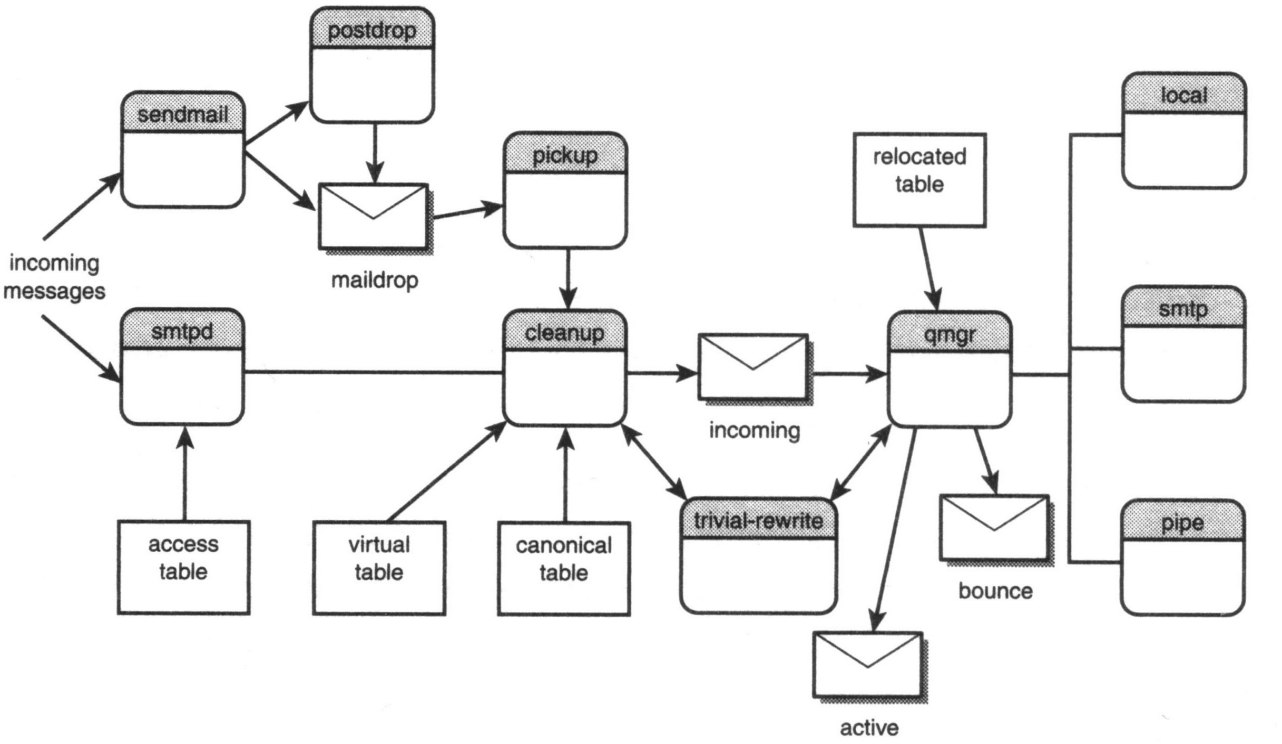
Warum Postfix?
Der meist verbreiteten Mail-Server in der Unix/Linux Welt ist `Sendmail`. Seit die Einführung von Sendmail, haben sich Mail-Administratoren mit der schwierige Konfiguration von Sendmail der Kopf zerbrochen weil sie so schwierig ist. Sendmail ist ein altes Konzept das als ein einziges grosses Programm läuft, dadurch ist sendmail nicht sehr schnell, und sendmail hat in der Vergangenheit öfter Sicherheitslöcher gehabt, die aber immer sehr schnell repariert worden sind. Die Erwähnten Eigenschaften von Sendmail motivieren sehr nach Alternativen zu suchen. Es gibt inzwischen viele Alternativen zu Sendmail (`http://www.sendmail.org` & .com):

- `Qmail` sehr schnell, sicher, flexibel, eigenes Mailbox-Format.
  `http://www.qmail.org`
- `Postfix` schnell, sicher, 120% kompatibel zu Sendmail.
  `http://www.postfix.org`
- `ZMailer` schnell, sicher, für sehr grosse Belastung geeignet.
  `http://www.zmailer.org`
- `Exim` klein und einfach zu konfigurieren, gute spam-Filters.
  `http://www.exim.org`
- `CommuniGate Pro`
  kommerzielles Produkt (ab $500), leichte Konfiguration über einen Browser, in der Mac-Welt verbreitet.
  `http://www.stalker.com/communigatepro`

Wir haben uns für Postfix entschieden, weil er gute Referenzen hat, einfach zu konfigurieren ist, kompatibel zu Sendmail ist und als RPM (mindestens bei SuSE) verfügbar ist. Konkret bedeutet dass, das es schnell möglich ist, Tests durchzuführen und zu positive Resultaten kommen.

- **Postifix-Information**
  - Literatur: Postfix von Richard Blum, Verlag: Sams (in englisch)
  - Internet: `http://www.postfix.org` (sehr viel Dokumentation)
- **Zusätzliche Dokumentation**
  - Amavis - **A MA**il **VI**rus **S**canner. `http://www.amavis.org`

`Postfix-Aufbau (page 35)`



- **<u>Mail processing sequence of events:</u>**
  - Receiving e-mail

From local user:

The <u>Local MUA</u> of local user uses `sendmail` to pass-on messages to the `maildrop` message queue: `/var/spool/postfix/maildrop/`*`codedmailname`*
Note: The local MUA `mail` uses also the `sendmail` program to process the mail.

The program `postdrop` is used automatically when the `maildrop` directory is not world writable. This is to restrict the write access of the directory to `postdrop`.

-The `maildrop` directory must be writable only from the group `maildrop` and `chmod 1730`.
- `postdrop` must be set SGID and owned by `postfix`, group `maildrop`.

The message waits in the `maildrop` directory until the `pickup` program takes it and forwards it to the `cleanup` program.

- **From remote host:**

  The Remote MUA communicates through the `smtpd` program using SMTP protocol.
  The `smtpd` uses the access table to verify the access rights of the remote host.
  The `smtpd` sends the message to `cleanup` program.

- **Mail Header Format (RFC 822)checking and cleanup by `cleanup` program.**
  Message header is checked against:

  - Missing `From: , Message-ID: , Date:`
  - Getting `To: , Cc: Bcc:` addresses
  - Checking for Addresses to rewrite against `canonical` and `virtual` tables
  - If header is invalid, then message is thrown away in the `corrupt` message queue

  FQDN Addresses Checking and rewriting:
  If header addresses are not FQDN the program `trivial-rewrite` converts it to FQDN:

  ```
  - user@host ------> user@host.domain
  - host!user ------> user@host.domain
  - user%domain ----> user@host.domain
  - user@site. -----> user@site
  ```

  -The `cleanup` program then puts the message in the `incoming` message queue.
  They are waiting there for `qmgr` program to process them.

- **Message processing and Delivery**
  - The program `qmgr` puts the message in the `active` message queue for processing(Study)
  - **Message processing with `qmgr` program**
    - If msg destination = local user, `local` program delivers it to local user mailbox.
      It checks `aliases` table and `~/.forward` file before delivery.
      The message can also be sent to `procmail` (external program)to deliver the local message. `~/.forward` file is only to forwarding to other local users.
    - If msg destination = remote server,
      `smtp` program attempt to deliver the message.
      - Undeliverable messages are logged in the `defer` directory and put in `deferred` message queue with a time stamp for retry delay.
        They will be tried again later.
      - Refused messages by remote mail server are forwarded to `bounce` program, processed (changed)and put in `bounce` message queue.
        They will be sent back to sender later by putting them in the `incoming` message queue.
      - Messages with unrecognizable addresses are sent to program `trivial-rewrite` converts it to FQDN before attempt sending:
    - Messages for other mail systems on same mail server are forwarded via the `pipe` program. eg. UUCP software.
    - Corrupted messages are saved in the corrupt message queue.
      Will be clean-up later.

- **Postfix-Interne-Programme**
  - `master` Main Postfix Daemon that controls the scheduling and the start and stop of the following internal programs of Postfix Mailing System. It is located in: `/usr/lib/postfix/master`
  - `bounce` Returns a bounced message to the sender and writes a log message in the `bounce` message queue Bounced messages can happen because local user doesn't exist or remote mail server not available.
  - `cleanup` Processes incoming mail Headers and places messages in the incoming queue.
  - `error` Processes messages delivery requests from `qmgr` program , forcing messages to bounce.
  - `local` Delivers Messages destined for local users.
  - `pickup` Waits for messages in the `maildrop` queue and sends them to the `cleanup` program to begin processing.
  - `pipe` Forwards messages from `qmgr` to other non-postfix programs.
  - `postdrop` Moves an local incoming message to the `maildrop` queue when that queue directory (`/var/spool/postfix/maildrop`)is not world writable.
  - `qmgr` Processes messages in the `incoming` queue, determining where and how they should be delivered, and spawns programs to deliver them. It manages the following queues: `incoming`, `active`, `deferred`, `corrupt`. And keeps an eye on the `bounce` and `defer` messages directories.
  - `smtp` SMTP Client that forwards messages to external mail servers.
  - `smtpd` SMTP Server that receives mail messages from external mail clients
  - `trivial-rewrite` Receive messages from `cleanup` to ensure the header addresses are in standard format for the `qmgr` program. Also used by the `qmgr` program to resolve remote addresses.
  - `showq` Reports Postfix mail queue status
  - `tlsmgr` Postfix TLS session cache and PRNG handling manager. For Secure Mailler using OpenSSL
  - `flush` Postfix fast flush server. This program expects to be run from the `master(8)` process manager. `man 8 flush` for more info. Location of "fast flush" logfiles `/var/spool/postfix/flush`

- **Postfix-Warteschlangen**
  - `maildrop` New messages waiting to be processed, received from local processes.
  - `incoming` New messages waiting to be processed, received from remote hosts as well as processed messages from local users.
  - `active` Messages that are ready to be delivered to `qmgr` program.
  - `defer` Log files of deferred mail messages
  - `deferred` Messages that have failed on an initial delivery attempt and are waiting for another attempt.
  - `mail` Delivered messages stored for local users to read.
- **Postfix-Werkzeuge**
  - `mailq` oder `sendmail -bp` zeigt die in der Warteschlange sind, die noch nicht ausgeliefert worden sind oder nicht ausgeliefert worden konnten.
  - `postfix flush` oder `sendmail -q` versucht alle Mails die in der Warteschlange sind, zu senden.
  - `postfix start (or stop, reload, abort, flush, or check)`
  - `postconf -n` zeigt die parameter die verändert worden sind.
  - `postconf -m` zeigt mit welchen Modulen Postfix kompiliert ist.
  - `newaliases` *`aliasdatei`* erstellt eine neue aliases-Datenbank
  - `postalias` Queries database for keywords and their values
  - `postcat` zeigt ein Mail von einer Warteschlange in "menschlicher Form" an.

    Beispiel:

    ```
    mailq
    find /var/spool/postfix/deferred -name XXXXXXXXX
    postcat /var/spool/postfix/deferred/x/y/XXXXXXXXX
    ```
  - `postlog` Allows to log a text line in the mail log file.
    Acts like `logger` program but just for `mail.*`
    eg. `postlog -i -p info -t` *`title Message`*
  - `postmap` */etc/postfix/mapfile*
    Converts text file to a database
  - `postsuper` Deletes or requeues messages in queues.
    Can only be executed by the superuser (root)
    eg. `postsuper -d ALL deferred`
    Deletes all messages of deferred queue
  - `postkick` Allows to send request to the specified service over a local postfix transport channel from external programs like shell scripts.
  - `postlock` Locks mail folder before executing a command
- **Extra tools not included in Postfix:**
  - `procmail` Powerful local mail delivery agent
    `formail` Re-formats/modifies mail headers
    `biff` Announces when a mail has arrived
- **Postfix-Lookup-Tabellen**

Lookup table	used by program	Description
access	smtpd	Accept/reject incoming mail according to source addr
aliases	local	Redirect mail coming in for local recipients
canonical	cleanup	Local and non local addresses mappings
relocated	qmgr	Info used to send notice back to sender for bounced messages
transport	trivial-rewrite	Mapping of destination domain to delivery methods
virtual	cleanup	Redirection of local and non-local recipients

- access Maps remote SMTP hosts to an accept/deny table for security according to sender name , domain, etc

  File Syntax Format: /etc/postfix/access (page 202)

```
roland@spamit.de          REJECT
sexygirl@broadband.sk.uk 554 No entrance permitted
marty@                    REJECT
linux.local               554  Not permitted
217.224                   REJECT   (not working yet !!!)
```

  Note: line starting with at least one space are continuation of previous line.

  IMPORTANT: Do not use tabs, use spaces between parameters

  Compile the table to hash database:

```
postmap /etc/postfix/access
```

  Declare the table in /etc/postfix/main.cf

```
smtpd_sender_restrictions = hash:/etc/postfix/access
```

- aliases (page 205) Maps alternative fictive local recipients to:
  - local users mailboxes
  - remote e-mail addresses
  - a local file
    in main.cf : allow_mail_to_files = yes
  - a local program via unnamed pipes
    in main.cf : allow_mail_to_commands = yes
  - multiple e-mail addresses via :include:/mailing/list/file

  other aliases main.cf entries:
  - alias_database hash:*/text_filename* (creates a .db file database)

  or - alias_database dbm:*/text_filename* (creates a .dbm file database)

  Text Format:(compatible with sendmail aliasses)

```
admin:   michel, michel@dozlinux.local, michel@mmbisson.com
admin2:  /tmp/vacation-mail.txt
test:    |/usr/bin/sendfax -n -d 5551212
savetxt: :include:/home/hans/mailing-list.txt
```

  Compile the table to hash database:

```
newaliases /etc/aliases
```

  Declare the table in /etc/postfix/main.cf

```
alias_maps = hash:/etc/aliase
```

- recipient_canonical and sender_canonicall (page 208) Maps alternative mailboxes to real mailboxes for rewriting sending and receiving messages headers.
  Used by cleanup program to rewrite addresses in the mail header.
  Good example:

In combination with aliases it allows to use long names
eg. michel.bisson@mymailserver.de to mean
michel@mymailserver.de
That would involve writing the following:
in aliases----> michel.bisson: michel
in sender_canonical--> michel michel.bisson

eg. To exchange only the sender address from an email:
in sender_canonical:
farbey@linuxint.com = joe.farbey@linuxint.com

Text Format:

LocalUserName long.email.name
eg. michel michel.bisson

Compile the table to hash database:
postmap /etc/postfix/sender_canonical
postmap /etc/postfix/recipient_canonical

Declare the table in /etc/postfix/main.cf
sender_canonical_maps= hash:/etc/postfix/sender_canonical
recipient_canonical_maps = hash:/etc/postfix/recipient_canonical

- relocated (page 209) Maps no longer valid user mailboxes (for bounced messages) to text inserted in bounced messages. The text insert can be anything. New name, address, street etc. The inserted text will follow a fixed message:

  user has moved to <Text inserted>

File Format: michel michel@newcompany.de Please change it.

Compile the table to hash database:
postmap /etc/postfix/relocated

Declare the table in /etc/postfix/main.cf
relocated_maps= hash:/etc/postfix/relocated

- transport (page 212) Maps Domain Names to delivery methods for remote hosts connectivity and delivery: local, uucp or smtp
  Can be used to specify a relay mail server which will forward to destination.

  File Format:

```
destination.domain transport:[nexthop][:port]
laptop.linux.local   local:   (needed for local server)
localhost.linux.local  local:
company.de           smtp:viaserver.de:8025
mmbisson.de          smtp:
special.com          uucp:
```

Compile the table to hash database:
postmap /etc/postfix/transport

Declare the table in /etc/postfix/main.cf
transport_maps= hash:/etc/postfix/transport
default_transport = smtp

- virtual (page 214)Maps recipients and domains to local mailboxes for delivery
  File Format:

linuxint.org virtual
considers all mail for linuxint.org as local mail

`michel michel@mmbisson.com michel@dozlinux.linux.local`
forward mail destined to local michel to another address
`martin@virtualmail.com mary`
forward all mail of martin to local user mary
`@linuxint.homelinux.com pierre@sun.linux.local`
forward all mail of one domain to a user in another domain

`pierre@globeall.dyndns.org michel@sun.linux.local`
forward mail of one address to another address

- **Compile the table to hash database:**
  `postmap /etc/postfix/virtua`
- **Declare the table in `/etc/postfix/main.cf`**
  `virtual_maps = hash:/etc/postfix/virtual`

- **Relaying mail.**
Postfix will accept to relay mail if the following conditions are met:
- If the mail's destination is a local mailbox
- If the sender is a local user (user logged-in in the host where postfix resides)
- If the following directives in `/etc/postfix/mail.cf` allows it like:

```
mynetworks = 127.0.0.1, 10.1.1.0/24
smtpd_recipient_restrictions =
permit_mynetworks, reject_unauth_destination
```

In this example postfix will relay mails that are sent from the mail clients programs residing inside the local network(10.1.1.0/24) and the localhost (127.0.0.1) and reject all other mails.

- **Postix Directories and files (für SuSE)**

`/etc/postfix/master.cf`	Postfix Daemon configuration for running core internal programs
`/etc/postfix/main.cf`	Configuration used by core programs to process messages.
`/etc/aliases`	Text database file of local users aliases
`/etc/aliases.db`	hash database file of local users aliases
`/etc/postfix/access`	
`/etc/postfix/access.db`	
`/etc/postfix/canonical`	
`/etc/postfix/canonical.db`	
`/etc/postfix/transport`	
`/etc/postfix/transport.db`	
`/etc/postfix/relocated`	
`/etc/postfix/relocated.db`	
`/etc/postfix/virtual`	
`/etc/postfix/virtual.db`	
`/etc/postfix/sender_canonical`	
`/etc/postfix/sender_canonical.db`	
`/etc/postfix/pcre_table`	
`/var/spool/mail/*`	Location of local users mailboxes
`/var/spool/postfix`	Message queues of postfix mail system
`/etc/postfix/postfix-script`	
`/etc/postfix/postfix-script-nosgid`	
`/etc/postfix/postfix-script-sgid`	
`/etc/postfix/regexp_table`	
`/etc/postfix/sample-aliases.cf`	Examples of configurations of `main.cf`.

```
/etc/postfix/sample-auth.cf
/etc/postfix/sample-canonical.cf
/etc/postfix/sample-compatibility.cf
/etc/postfix/sample-debug.cf
/etc/postfix/sample-filter.cf
/etc/postfix/sample-flush.cf
/etc/postfix/sample-ldap.cf
/etc/postfix/sample-lmtp.cf
/etc/postfix/sample-local.cf
/etc/postfix/sample-misc.cf
/etc/postfix/sample-pcre.cf
/etc/postfix/sample-rate.cf
/etc/postfix/sample-regexp.cf
/etc/postfix/sample-relocated.cf
/etc/postfix/sample-resource.cf
/etc/postfix/sample-rewrite.cf
/etc/postfix/sample-smtp.cf
/etc/postfix/sample-smtpd.cf
/etc/postfix/sample-tls.cf
/etc/postfix/sample-transport.cf
/etc/postfix/sample-virtual.cf
/etc/permissions.d/postfix
```

`/etc/init.d/postfix`	SuSE Script to start/stop Postfix run level service
`/sbin/rcpostfix`	SuSE Symbolic Link to above `/etc/init.d/postfix`
`/var/log/mail`	Log file for all mail transactions
`/var/mail/`	Symbolic link to `/var/spool/mail/`

---------- Postfix mail system Core programs ----------

Note: These programs are only started by master daemon or other core programs

`/usr/lib/postfix/bounce`	Rewrites and Bounces e-mails
`/usr/lib/postfix/cleanup`	Checks and rewrites message headers
`/usr/lib/postfix/error`	Handles problematic message delivery
`/usr/lib/postfix/flush`	Postfix fast flush server
`/usr/lib/postfix/lmtp`	Handles the lmtp protocol connections
`/usr/lib/postfix/local`	Delivers local e-mails in mailboxes
`/usr/lib/postfix/master`	Main daemon controlling core programs
`/usr/lib/postfix/pickup`	Transfers mails from `maildrop` message queue to cleanup program.
`/usr/lib/postfix/pipe`	Passes mails to external programs
`/usr/lib/postfix/qmgr`	before delivery mail queue manager
`/usr/lib/postfix/showq`	Informs programs about messages queues
`/usr/lib/postfix/smtp`	Sends mails to mail servers using smtp protocol
`/usr/lib/postfix/smtpd`	Receives mail from hosts using smtp protocol
`/usr/lib/postfix/trivial-rewrite`	Rewrites headers to ensure FQDN
`/usr/lib/postfix/spawn`	daemon provides the Postfix equivalent of `inetd`
`/usr/lib/postfix/tlsmgr`	Manages TLS secure smtp connections if used

---------- Postfix Tools ----------

`/usr/bin/mailq`	Shows the curent mail queue
`/usr/bin/newaliases`	Translates text (sendmail) aliases to databases
`/usr/sbin/postalias`	Queries and modifies the postfix aliases database eg. `postalias -q mail /etc/aliases`
`/usr/sbin/postfix`	Main postfix program (controls `master`)
`/usr/sbin/sendmail`	`Sendmail` like Postfix compatible interface
`/usr/lib/sendmail`	Symbolic link to above `/usr/sbin/sendmail`
`/usr/sbin/postcat`	Displays the content of a message in a queue
`/usr/sbin/postconf`	Displays configurations entries in `main.cf`

`/usr/sbin/postdrop`	Program used to deposit messages in the `maildrop` message queue if `maildrop` is not world readable.
`/usr/sbin/postkick`	Allows to send request to the specified service over a local postfix transport channel from external programs like shell scripts.
`/usr/sbin/postlock`	locks mail folder before executing a command
`/usr/sbin/postlog`	Allows to log a text line in the mail log file. Acts like `logger` program but just for `mail.*` eg. `postlog -i -p info -t` *`title Message`*
`/usr/sbin/postmap`	Converts text lookup tables to databases. (`xx.db`)
`/usr/sbin/postsuper`	Deletes or requeues messages in queues. eg. `postsuper -d ALL deferred` Deletes all messages of defered queue eg.2 `postsuper -d` *`MailID`* *`Mail-ID`*= Mail ID from `mailq` command.
`/usr/sbin/qshape [incoming|active|deferred|hold]`	Displays the number of mails in a particular queue. incoming, active, deferred or hold Under the title 'T' is the total for that queue.
`/usr/sbin/smtp-sink`	???
`/usr/sbin/smtp-source`	???
`/var/adm/fillup-templates/rc.config.d.postfix`	???
`/var/adm/fillup-templates/rc.config.postfix`	???

(SuSE)-Postfix Fehlerbehebung

- Der "Einfluss" von SuSE auf Postfix kann ausgeschaltet werden: mit YaST die Variable **`POSTFIX_CREATECF = no`** setzten
- **`Achtung!`** SuSE definiert die Postfix-Parameters am Ende der Datei `main.cf`.
- SuSE 7.3 hat schon eine Aktualisierung von `postfix.rpm` herausgegeben die nicht ganz in Ordnung war. `postdrop` funktionierte nicht mehr. Das Programm `/usr/sbin/postdrop` soll so aussehen:

  ```
  -rwxr-sr-x  1 root  maildrop  80523 Dec 12 10:22 /usr/sbin/postdrop
  ```

- Das erste Mal wenn Postfix gestartet wird, ist es interessant die Protokolldatei `/var/log/mail` anzuschauen, um zu kontrollieren ob alles in Ordnung Hochfährt. Es ist schon passiert das die Aliases-Dantenbank (`aliases.db`) irgendwie nicht lesbar ist. Diese Problem lässt sich leicht beheben indem `newaliases` Befehl aufgerufen wird und Postfix neu gestartet wird. (`rcpostfix reload`). Wenn eine andere Lookup-Tabelle beim ersten starten nicht lesbar ist, kann die Tabelle mit `postmap hash:/etc/postfix/Tabelle` neu gemacht werden. Danach muss `postfix` wieder neu gestartet werden.
- **Alle Mails in den Warteschlangen löschen:**

  ```
  find /var/spool/postfix/deferred -type f -exec rm {} \;
  find /var/spool/postfix/defer    -type f -exec rm {} \;
  ```

- **MIME Mail encoding:**

  Example of Mail header including MIME

  ```
  sendmail michel.dozlinux.local
  Subject: hallo in html
  Mime-Version: 1.0
  Content-type: text/html
  <Body><H1><Font color=red>
        hallo world
  ```

```
</Font></H1></body>
```

- **Einige Postfix-Parametern in main.cf**

`myhostname`	Rechnername + Domäne des Rechners auf dem Postfix läuft.
`mydestination`	Rechnernamen und/oder Domäne die Postfix als End-Station sieht. List of domains that this mail system considers as local.
`myorigin`	Domäne die am Sender des Emails angehängt wird. Sehr praktisch mit virtuelle Domäne oder wenn Postfix auf einem Rechner läuft der keine wirkliche Internet-Domäne besitzt.

`defer_transport = smtp`
Die Mails werden in der Warteschlangehereingesetzt und werden nach dem Befehl `postfix flush` gesendet. Das ist für "dial up" Verbindungen praktisch.

`mail_name =` Zeichenkette das Postfix herausgibt wenn er auf dem Port 25 angefragt wird (banner).

`inet_interfaces = 127.0.0.1` (und noch `eth`*x* IP Nummern)

Mail automatisch abholen mit fetchmail

- `fetchmail` holt Mails über POP3 oder IMAP, und gibt sie weiter über `smtp` am lokalen Mail-Server (Postfix, qmail, Sendmail usw.). Wenn es keinen lokalen Mail-Server gibt, dann gibt `fetchmail` die Mail an eine MDA wie z.B. `procmail` weiter.
- Unter SuSE befindet sich das Paket `fetchmail` in SuSE CD.
- **Konfigurationsdateien von fetchmail:**

  `/etc/fetchmailrc` heissen, oder `/root/.fetchmailrc`.
  Diese Datei muss erstellt werden mit den Zugriffsrechten `600`.
  Machen Sie sicher dass der Benutzer `fetchmail` hatte `/bin/sh` oder `/bin/bash` als shell.

  Noch eine Konfigurationsdatei unter SuSE ist: `/etc/sysconfig/fetchmail`
  z.B. Fetchmail interval settings und andere sind da.

  Example of the configuration file: `/etc/fetchmailrc`

```
defaults protocol pop3
set daemon 300 (sets the fetch interval to 300 sec.(5 Min)
poll "pop.tiscalinet.de"
     user "john-Martin" with password "passwort" is john here;
poll "mail.tiscali-dsl.de" protocol pop3
     user "benutzername" with password "passwort" is joe here;
poll "post.strato.de" (Note:the usenames include domains at strato.de)
     user "linux@globeall.de" with password "passwort" is
     pierre here;
     user "info@linuxint.de" with password "passwort" is
     michel here mda "/usr/sbin/sendmail -oem -f %F %T";
```

- To control (start/stop/status) fetchmail daemon:
  Important: If you used fetchmailconf to configure it then copy `/root/.fetchmailrc` to `/etc/fetchmailrc`

```
rcfetchmail { start | stop | restart | reload | status }
/etc/init.s/fetchmail ""       ""       ""       ""
```

- To insert fetchmail in default runlevel:
  `insserv fetchmail`
- Fetchmail kann in `/etc/ppp/ip-up.local` eingefügt werden:
  `/etc/init.d/fetchmail start`
- und in `/etc/ppp/ip-down.local`:
  `/etc/init.d/fetchmail stop`
- natürlich kann fetchmail auch direkt als Befehl ausgeführt werden:

```
/usr/bin/fetchmail   -d 120 -a -f /etc/fetchmailrc \
                     -L /var/log/fetchmail

-d    startet fetchmail als Dämon, alle 120 sec
-a    holt alle Mails, die alten und neuen
-f    Konfigurationsdatei von fetchmail
-L    Logfile

/usr/bin/fetchmail -quit            (stops fetchmail)
```

- Documentation:
  A lot of documentation is available after installation in:
  `/usr/share/doc/packages/fetchmail`
- **Fetchmailconf**
  This program is a graphic interface program that helps to configure fetchmail, to test it temporarily and to make it ready for permanent work.

  - **Installation:** Package: `fetchmailconf` from SuSE CD
  - **Starting Fetchmailconf**
    Since Fetchmailconf makes changes to the system's configuration, it must be started as root user to be allowed to save the changes.
    `kdesu fetchmailconf`
  - **Using Fetchmailconf:**
    - Click on the button 'Configure Fetchmail' to get to the configuration window
    - Click on 'Novice Configuration'
    - In the 2nd window:
      - Enter the Interval(in minutes) between mail fetching events.
      - Enter the POP3 or or IMAP servername and press <Enter>
    - In the 3rd window:
      - Select the type of mail protocol to fetch the mail (eg. POP3)
      - Enter the remote username for Authentication on the remote server and press <Enter>
    - In the 4th window:
      - Enter the user's password
      - (Optional) Enter the SSL configuration parameters.
      - Select the local username to where the fetched mails should be delivered.
      - Click on OK
    - In the 3rd window:
      - Click on OK
    - In the 2nd window:
      - Click on 'Save'
      - Click on yes to agree to overwrite the original configuration file.

Configuration file `/root/.fetchmailrc` will be written.

- On the 1st window:
  - Click on top 'Run Fetchmail' for testing it first.
  Fetchmail will run and fetch the mailbox on the server and save it in the local user's mailbox. Check the new mail in the local mailbox:
  `mail`

- ## Mail-Zugang über POP3 und IMAP zuverfügung stellen
  - Den nächste Schritt ist der Zugang zu den Mailboxes auf dem lokalen Mailserver von Klienten zu erlauben.
  - Für POP3 gibt es diePaket `imap` von BSD (Dämon `ipop3d`) und `qpopper` (Dämon = `popper`), das von `Qualcomm` gepflegt wird .
  - Für IMAP ist auch das Paket `imap` zuständig (Dämon `imapd`).
  - Alle diese Dämonen können über der `inetd` gestartet werden:

    Datei `/etc/inetd.conf`:

```
#pop3 stream tcp nowait root /usr/sbin/tcpd /usr/sbin/popper -s
pop3 stream tcp nowait root /usr/sbin/tcpd /usr/sbin/ipop3d
imap stream tcp nowait root /usr/sbin/tcpd /usr/sbin/imapd
```

    Nach einer Änderung in der Datei `/etc/inetd.conf` muss der Dämon `inetd` neu gestartet werden (`rcinetd reload` oder `killall -HUP inetd`)
  - Mehr muss nicht gemacht werden. Von einem Klienten, können jetzt die Mails über POP3 oder IMAP geholt werden. Der Benutzname und das Passwort sind die vom Benutzer-Konto des Rechners aufdem der Mail-Server läuft.
  - IMAP server automatically pics-up mail from each user mailbox(`/var/mail/user`) when the user is connecting and transfers it to `~/mbox`. It then reads the mbox and works on it. Reading , deleting and new mail is all done in the `~/mbox.`

  - **IMPORTANT: POP3 Passwords are NOT secure!**
    If you install the programm '`dsniff`' and run the command:
    `dsniff -m -i eth0`
    and connect from `kmail` to a pop3 server or someone connect to the local pop3 server, then the name and password will be seen in the dsniff terminal.!!!
    Solution: install the **pop3s** server that follows

  - **To check the POP3 mail on a remote host using 'mail':**
    `mail -f` show the local mbox's content of the current user, then issue the command:
    `folder pop3://user@popmailserver.com`
    Give password and then issue the command:
    `headers`
    to see the list of currently waiting mails in mailbox.

POP3S (Secure pop3) Configuration

- Install the package '`imap`'
- Run the commands:
  `cd /etc/ssl/certs`
  `openssl req -new -x509 -nodes -out ipop3d.pem -keyout ipop3d.pem`
  Answer the questions(can be anything)
- Edit the file `/etc/xinetd.d/imap`
  Under the section '`service pop3s`'
  `disable = no`
- Run the command `rcxinetd restart`
- In the Mail client pop configuration, use SSL and `Plain` Login method.
  Enter the user login name and password.

- **Secure SMTP with SASL(SuSE 9.2/10.x)**
- **Installation:**
  Install the following packages:
  `cyrus-sasl, cyrus-sasl-crammd5, cyrus-sasl-digestmd5`
  `cyrus-sasl-saslauthd ,cyrus-sasl-plain`
- **Postfix basic configuration:**
  in `/etc/postfix/main.cf`
  Make sure that following 2 parameters are entered properly:
  `inet_interfaces = 127.0.0.1 ::1` ***<HostIP>***
  `myhostname =` *<Hostname>*

  eg. `inet_interfaces = 127.0.0.1 ::1 192.168.100.70`
  `myhostname = laptop.linux.site`
- **To activate `sasl` authentication do the following:**
  in `/etc/postfix/main.cf`

```
broken_sasl_auth_clients = yes
smtpd_sasl_auth_enable = yes
smtpd_sasl_application_name = smtpd
smtpd_sasl_local_domain = $myhostname
smtpd_recipient_restrictions =
        permit_mynetworks,
        permit_sasl_authenticated,
        reject
```

- **To use `/etc/sasldb2` database for passwords:**
  - Make sure that the group `postfix` can have read access to `/etc/sasldb2`

```
chown root.postfix /etc/sasldb2
chmod 640 /etc/sasldb2
```

  - In `/usr/lib/sasl2/smtpd.conf`:

```
pwcheck_method: auxprop
auxprop_plugin: sasldb
mech_list: plain login
```

  - To create a new `/etc/sasldb2` user:

```
saslpasswd2 -c -u $(postconf -h myhostname) username
```

  eg. `saslpasswd2 -c -u $(postconf -h myhostname) michel`

  - To delete a user from `/etc/sasldb2` :

```
saslpasswd2 -d username
```

  - To list the sasl users and their realms from `/etc/sasldb2` password database:

```
sasldblistusers2
```

- **To use the server's shadow password system via PAM:**
  - Start the `saslauthd` Daemon:

```
rcsaslauthd start
insserv saslauthd
```

  (for permanent start at boot time)

  - In `/usr/lib/sasl2/smtpd.conf`:

```
pwcheck_method: saslauthd
mech_list: plain login cram-md5
```

Or:

- Using the `sasl` authentication method instead of PAM

```
pwcheck_method: auxprop
auxprop_plugin: sasldb
mech_list: plain login cram-md5
```

- To add new users to sasl authentication:

```
mkdir /etc/empty
useradd -mk /etc/empty -s /bin/false username
```

- To test locally the `sasl` authentication:

```
testsaslauthd -u username -p password
```

- **MAIL CLIENT configuration:**
  - Port 25
  - Need authentication(Give name and password)
  - Encryption=NONE
  - Authentication=LOGIN

- More info in:
  `/usr/share/doc/packages/postfix/README_FILES/SASL_README`

- ## Forward und Vacation Funktionen

The file `~/.forward`
will activate the forwarding of the user's mail to another local user.
Just enter the local username of the user to which the mail should be forwarded.

- **Protecting mail against virusses/spam with amavis-new**(Suse 9.2/9.3)

1) INSTALLATION
Install the following packages from SuSE 9.2/9.3 distribution Cds/DVD:
- `postfix`
- `amavis-new`
- `clamav`
- `clamav-db`(only if you don't update the virus signatures database from Internet)
- `antivir`
- `antivir-avguard` (on SuSE 10.1)
- `perl-spamassassin`
- `spamassassin`

2) CONFIGURATION:

- **AMAVIS**
  - Edit the file `/etc/amavisd.conf`
    Adapt the follwoing line: (around line 18) to be the FQDN of the local host
    eg. `$mydomain = 'laptop.linux.site';`
    Amavis will send an email to root user of this above host per refused mail.
- **ANTIVIR**
  - Edit the file: `/etc/antivir.conf` and change the email address for virus notification: eg. `EmailTo root@laptop.linux.site`

  - Make sure the `dakuso` kernel module is loaded at boot time:
    add `dazuko` to the `MODULES_LOADED_ON_BOOT` variable
    `/etc/sysconfig/kernel` before the capability module, e.g.:
    `MODULES_LOADED_ON_BOOT="dazuko capability"`
  (optional)You can manually prepare the system now for testing by doing:
  ```
  rmmod capability
  modprobe dazuko
  modprobe capability
  ```
- **CLAMAV**
  (Optional)Edit the configuration file: `/etc/freshclam.conf`
  It can be edited to change the frequency per day of the database updating:
  eg. `Checks 12 (Default)`
  (Updates the virus signatures database 12 times a day)
  Run the command `freshclam` if you're connected to the internet to get the latest virus signatures database. Later `freshclam` will be run automatically from clamav.

- **SPAMASSASSIN**
  Nothing to do.
- **SOPHOS Virus scanner**
  - Get the latest version of the Sophos(Linux on Intel using libc6 (glibc2.2) at:
    http://www.sophos.com/support/updates/sophos-anti-virus-non-windows.html
  - Unpack the Sophos tarball file in `/usr/local/Sophos-Install`
  - Do the following commands:
    ```
    cd /usr/local/Sophos-Install
    ./install.sh
    ```
  - Uncomment the Sophos Virus scanner lines at the end of `/etc/amavis.conf`

- **POSTFIX**
  Use Yast to configure the use of Amavis Virus scanner (cross the appropriate box)
  or edit the file: `/etc/postfix/master.cf` and change the following first line from:

```
smtp  inet  n  -  n  -  2  smtpd
```

  to

```
smtp  inet  n  -  n  -  2  smtpd -o content_filter=smtp:[127.0.0.1]:10024
```

  and add the following line:

```
localhost:10025 inet  n  -  n  -  -  smtpd -o content_filter=
```

- **Starting sequence:**

  Postfix Service: `rcpostfix start`
  AntiVir Daemon: `rcavguard start`
  ClamAV Daemon: `rcclamd start`
  Spamd Daemon: `rcspamd start`
  AmaVis Daemon: `rcamavis start`
  ClamAV DB Update: `rcfreshclam start`

  To make sure they all start at boot time:

```
insserv postfix avguard clamd amavis freshclam spamd
```

- **More INFO on Virus scanners**

  - **AMAVIS** (TCP Port 10024)
    The Virus notification mail will be sent to the root user of this defined host.
    The virus mails will be quarantained into the directory defined by the following entry: `$QUARANTINEDIR = '/var/spool/amavis/virusmails';`
    The working directory of Amavis is defined by the following entry:

```
$MYHOME = '/var/spool/amavis';
```

    Optional:
    Disabling all mails virus checks and banned names:
    To prevent Virus/Banned/SPAM names checks on ALL incoming mails then insert the following directives:(In SuSE you only need to uncomment the lines.)

```
@bypass_virus_checks_maps = (1);
@bypass_spam_checks_maps  = (1);
```

    If you want to prevent Virus checks on mails for certain recipients, then here are some examples of filters(in `/etc/amavis.conf`) that do that. Note here that the virus and banned checks are separate to allow for finer filtering.

    Disabling all mails virus checks and banned names(for attached files) for the user `michel` for the domain `linux.site` and its subdomains.

```
@bypass_virus_checks_acl = qw( michel em .linux.site );
@bypass_banned_checks_acl = qw( michel em .linux.site );
```

    Disabling all mails virus checks and banned names(for attached files) for the domain `linux.site` but not for its subdomains.

```
@bypass_virus_checks_acl = qw( linux.site );
@bypass_banned_checks_acl = qw( linux.site );
```

    Sending all virus mails and banned mails to one recipient(virus administrator) for later checking.
    This feature involves a few steps:
    - Create the user `infected` in the system

```
useradd infected ; passwd infected
```

    - Include the following directives in `/etc/amavis.conf`

```
$virus_quarantine_to = 'infected@';
$banned_quarantine_to = 'infected@';
```

The user `infected` can now retrieve the infected mails like other mails and pick them up via the pop3 server.

- **CLAMAV** (TCP port 3310)
  Adapt the file: `/etc/clamd.conf` if needed. (normally not needed)
  Notification of virus check:
  The default is to send a syslog message as '`mail`' facility message.
  Normally it would be seen in `/var/log/mail` log file.
  Its virus database directory is `/var/lib/clamav`
  Its working TCP port is: `3310`
  Updating regularly ClamAV virus database:
  It is done by running the daemon `freshclam` with the command:

```
rcfreshclam start
```

- **ANTIVIR & AVGUARD**
  Antivir is composed of 2 Virus Scanners:
  - Access scanner: `antivir`
  - System Virus Scanner: `avguard`
    - works by loading a kernel module called: `dazuko`

  ANTIVIR:
  Adapt the file: `/etc/antivir.conf` and `/etc/avguard.conf` if needed.
  (Normally not needed) Its working directory is: `/usr/lib/Antivir`

  AVGUARD:
  If you want to use AvGuard, you have to disable at least the selinux framework, using the kernel boot parameter "`selinux=0`" and "`capability=0`".
  NOTE: remember that by disabling these modules, you will have trouble running `named` and `dhcpd` servers which need the '`capability`' module.
  Updating regularly the AntiVir Virus Database:
  - Create a cron job with the command: `/usr/bin/antivir -q --update`
  NOTE: The ANTIVIR license from SuSE doesn't allow for automatized updates.
  For more info read the file:
  `/usr/share/doc/packages/antivir/README.SuSE`

- **SPAMASSASSIN**
  [Optional]

  To make sure that spamassassin 'learns' further about what is a spam or ham(good mail) then do the following:
  - Create 2 spam user accounts in the mail server where spamassassin resides:

```
useradd -g nogroup -s /bin/false spamadmin
useradd -g nogroup -s /bin/false hamadmin
```

  - Make sure that the users in the network are forwarding:
  their non-tagged spam mails to spamadmin@*server.site*
  and their ***SPAM*** tagged good mails to hamadmin@*server.site*
  Note: Tagged mails are the ones that have already received the extra ***SPAM*** tag in the Subject field.

  - Run the following script regularly: (cron job)

```
#!/bin/bash
mkdir /var/spool/spam 2>/dev/null
mkdir /var/spool/oldspam 2>/dev/null
mkdir /var/spool/ham 2>/dev/null
```

```
mkdir /var/spool/oldham 2>/dev/null
mv /var/mail/spamadmin /var/spool/spam/spam_$(date -'+%Y.%m.%d-%H.%M.%S')
mv /var/mail/hamadmin  /var/spool/ham/ham_$(date -'+%Y.%m.%d-%H.%M.%S')
sa-learn --spam /var/spool/spam
sa-learn --ham  /var/spool/ham
mv /var/spool/ham/*  /var/spool/oldham
mv /var/spool/spam/* /var/spool/oldspam
```

NOTE: Make sure that the number of Spams and Hams mails given to the learner program is around the same. Learning only from spams mails doesn't work and can lead to many false recognitions.

- **SOPHOS Virus scanner**
- Installing Sophos:
  - Install `wget` in the system.(needed for the auto update of virus database)
  - Get the latest tarball from:

`http://www.sophos.com/support/updates/sophos-anti-virus-non-windows.html`

  - You need an EM Library name and password to download it.
    Make sure you get the right version for for you installed `glibc`.
    `Linux on Intel using libc6 (glibc2.2)` for SuSE 9.3
  - Extract the file in a directory like `/usr/local/Sophos-install`
  - Run the script `/usr/local/Sophos-install/install.sh`
  - Just run these commands once after the installation to make sure that the directory `/usr/local/ide` is a symbolic link to the latest installed ide's.

```
mv /usr/local/ide /usr/local/ide_1
ln -s /usr/local/ide_1 /usr/local/ide
```

  - Uncomment the lines pertaining to Sophos in `/etc/amavis.conf` (almost at the very end of the file). Then restart `amavis`. The `/var/log/mail` should show that `amavis` recognized the virus Sophos virus scanner.
  Note: (Optional)To make sure that Sophos is seen as a primary virus scanner, move the Sophos lines from the backup scanners section:

```
@av_scanners_backup = (.....
```

  to the primary scanners section:

```
@av_scanners = (.....
```

- A virus reporting program daemon(`icheckd`)is delivered with it.(optional)
  It receives virus reports from network clients sophos scanners and produces a report of viruses. To install and run it, run the script:
  `install -i` from the Sophos installation directory.
- The main virus scanner is: `sweep`. It is normally used by Amavis.

  The scanner program `sweep` can also be used manually:
  `sweep /` (Scans the whole system for viruses)
  `sweep /dir/to/my/file` (Scans a file for viruses)
  many other ways to use sweep are documented on the web site.
- The auto-update of the virus database is using a shell script and a perl script that are not part of the standard package. They are called:
  `/etc/cron.daily/Sophos.autoupdate` (shell script)
  `/usr/local/bin/Sophos_autoupdate` (perl script)
  `Sophos.autoupdate` is triggered daily by cron and it calls the perl script. Some parameters at the beginning of the perl script can be adjusted to match the current version of Sophos. It also needs the programs `wget` to be installed in the system. This script automatically retrieves the latest virus database from the Internet, `http://www.sophos.com/downloads/ide/`

saves it in a new directory (/usr/local/and changes the symbolic link: `/usr/local/ide` to point to this new directory.
A large database of older viruses is also located in a fixed location in `/usr/local/sav`

- **POSTFIX:**

  To send the virus notifications to another user than root then modify the file: `/etc/aliases` as follows:

  ```
  root: michel
  ```

  and run the command:

  ```
  newaliases
  ```

  NOTE: Watch the `/var/log/mail` while loading the AmaVis Daemon. It will display the name of the virus scanners it automatically finds and use, as well as other important information on what AmaVis uses to scan the mails.

- **Blocking SPAM via Internet 'Black list'**
  There are a few black lists servers on the Internet that can be used to block unwanted SPAM Mails. Postfix is already capable to use these blacklists. Here are the directives that need to be written in the `main.cf` configuration file from Postfix:

```
smtpd_client_restrictions =
      reject_rbl_client dul.dnsbl.sorbs.net
 or
      reject_rbl_client sbl-xbl.spamhaus.org
 or
      reject_rbl_client list.dsbl.org,
```

- Good example for mail filtering:

```
smtpd_recipient_restrictions =
   check_recipient_access hash:/etc/postfix/spam_rec_addr,
   permit_mynetworks,
   permit_sasl_authenticated,
   reject_unauth_destination,
   reject_invalid_hostname,
   reject_non_fqdn_sender,
   reject_non_fqdn_recipient,
   reject_unknown_sender_domain,
   reject_unknown_recipient_domain,
   reject_rbl_client blackholes.easynet.nl,
   reject_rbl_client cbl.abuseat.org,
   reject_rbl_client proxies.blackholes.wirehub.net,
   reject_rbl_client dnsbl.njabl.org,
   reject_rbl_client list.dsbl.org,
   reject_rbl_client sbl-xbl.spamhaus.org,
   reject_rbl_client bl.spamcop.net,
   reject_rhsbl_client blackhole.securitysage.com,
   reject_rhsbl_sender blackhole.securitysage.com,
   permit


smtpd_data_restrictions =
   reject_unauth_pipelining


smtpd_sender_restrictions =
   permit_mynetworks,
   permit_sasl_authenticated,
   reject_unknown_sender_domain,
   reject_non_fqdn_sender,
   check_sender_access hash:/etc/postfix/spam_addr
   permit
```

The following one rejects mails from Yahoo

```
#    reject_rbl_client bl.spamcop.net,
```

- **Controlling access/relay of postfix**
  Multiple directives in the `main.cf` file allow to restrict the postfix access.
  Here is a list of them and how they work:
  - The table below summarizes the purpose of each SMTP access restriction list. All lists use the exact same syntax; they differ only in the time of evaluation and in the effect of a REJECT or DEFER result.
  - Each restriction list is evaluated from left to right until some restriction produces a result of PERMIT, REJECT or DEFER (try again later). The end of the list is equivalent to a PERMIT result. By placing a PERMIT restriction before a REJECT restriction you can make exceptions for specific clients or users. This is called

`whitelisting`; the last example above allows mail from local networks but otherwise rejects mail to arbitrary destinations.

Restriction list name	Status	Effect of REJECT or DEFER result
`smtpd_client_restrictions`	Optional	Reject all client commands
`smtpd_helo_restrictions`	Optional	Reject HELO/EHLO information
`smtpd_sender_restrictions`	Optional	Reject MAIL FROM information
`smtpd_recipient_restrictions`	Required	Reject RCPT TO information
`smtpd_data_restrictions`	Optional	Reject DATA command
`smtpd_end_of_data_restrictions`	Optional	Reject END-OF-DATA command
`smtpd_etrn_restrictions`	Optional	Reject ETRN command

Examples:

```
# Allow connections from trusted networks only.
smtpd_client_restrictions = permit_mynetworks, reject

# Don't talk to mail systems that don't know their own hostname.
# With Postfix < 2.3, specify reject_unknown_hostname.
smtpd_helo_restrictions = reject_unknown_helo_hostname

# Don't accept mail from domains that don't exist.
smtpd_sender_restrictions = reject_unknown_sender_domain

# Block clients that speak too early.
smtpd_data_restrictions = reject_unauth_pipelining

# Enforce mail volume quota via policy service callouts.
smtpd_end_of_data_restrictions = check_policy_service
                                    unix:private/policy
# Whitelisting: local clients may specify any destination. Others may not.
smtpd_recipient_restrictions = permit_mynetworks,
                                  reject_unauth_destination
```

One powerful directive is the last one: `smtpd_recipient_restrictions`.
It allows to restrict the relaying of mails according to different rules.

smtpd_recipient_restrictions (default: permit_mynetworks, reject_unauth_destination)

The access restrictions that the Postfix SMTP server applies in the context of the RCPT TO command.

By default, the Postfix SMTP server accepts:

- Mail from clients whose IP address matches $mynetworks, or:
- Mail to remote destinations that match $relay_domains, except for addresses that contain sender-specified routing (user@elsewhere@domain), or:
- Mail to local destinations that match $inet_interfaces or $proxy_interfaces, $mydestination, $virtual_alias_domains, or $virtual_mailbox_domains.

IMPORTANT: If you change this parameter setting, you must specify at least one of the following restrictions. Otherwise Postfix will refuse to receive mail:

```
reject, defer, defer_if_permit, reject_unauth_destination
```

Specify a list of restrictions, separated by commas and/or whitespace. Continue long lines by starting the next line with whitespace. Restrictions are applied in the order as specified; the first restriction that matches wins.

The following restrictions are specific to the recipient address that is received with the RCPT TO command.

check_recipient_access *type:table*
Search the specified access(5) database for the resolved RCPT TO address, domain, parent domains, or localpart@, and execute the corresponding action.

check_recipient_mx_access *type:table*
Search the specified access(5) database for the MX hosts for the RCPT TO address, and execute the corresponding action. Note: a result of "OK" is not allowed for safety reasons. Instead, use DUNNO in order to exclude specific hosts from blacklists. This feature is available in Postfix 2.1 and later.

check_recipient_ns_access *type:table*
Search the specified access(5) database for the DNS servers for the RCPT TO address, and execute the corresponding action. Note: a result of "OK" is not allowed for safety reasons. Instead, use DUNNO in order to exclude specific hosts from blacklists. This feature is available in Postfix 2.1 and later.

permit_auth_destination
Permit the request when one of the following is true:
- Postfix is mail forwarder: the resolved RCPT TO address matches $relay_domains or a subdomain thereof, and the address contains no sender-specified routing (user@elsewhere@domain),
- Postfix is the final destination: the resolved RCPT TO address matches $mydestination, $inet_interfaces, $proxy_interfaces, $virtual_alias_domains, or $virtual_mailbox_domains, and the address contains no sender-specified routing (user@elsewhere@domain).

permit_mx_backup
Permit the request when the local mail system is MX host for the RCPT TO address. This includes the case that the local mail system is the final destination. However, the SMTP server will not forward mail with addresses that have sender-specified routing information (example: user@elsewhere@domain). Use the optional permit_mx_backup_networks parameter to require that the primary MX hosts match a list of network blocks.
Note: prior to Postfix version 2.0, use of permit_mx_backup is not recommended; mail may be rejected in case of a temporary DNS lookup problem.

reject_non_fqdn_recipient
Reject the request when the RCPT TO address is not in fully-qualified domain form, as required by the RFC.
The non_fqdn_reject_code parameter specifies the response code to rejected requests (default: 504).

reject_rhsbl_recipient *rbl_domain=d.d.d.d*
Reject the request when the RCPT TO domain is listed with the A record "*d.d.d.d*" under *rbl_domain* (Postfix version 2.1 and later only). If no "=*d.d.d.d*" is specified, reject the request when the reversed client network address is listed with any A record under *rbl_domain*.
The maps_rbl_reject_code parameter specifies the response code for rejected requests (default: 554); the default_rbl_reply parameter specifies the default server reply; and the rbl_reply_maps parameter specifies tables with server replies indexed by *rbl_domain*. This feature is available in Postfix 2.0 and later.

reject_unauth_destination
Reject the request unless one of the following is true:
- Postfix is mail forwarder: the resolved RCPT TO address matches $relay_domains or a subdomain thereof, and contains no sender-specified

routing (user@elsewhere@domain),
- Postfix is the final destination: the resolved RCPT TO address matches $mydestination, $inet_interfaces, $proxy_interfaces, $virtual_alias_domains, or $virtual_mailbox_domains, and contains no sender-specified routing (user@elsewhere@domain).

The relay_domains_reject_code parameter specifies the response code for rejected requests (default: 554).

reject_unknown_recipient_domain
Reject the request when the RCPT TO address has no DNS A or MX record and Postfix is not final destination for the recipient address.
The unknown_address_reject_code parameter specifies the response code for rejected requests (default: 450). The response is always 450 in case of a temporary DNS error.

reject_unlisted_recipient (with Postfix 2.0: check_recipient_maps)
Reject the request when the RCPT TO address is not listed in the list of valid recipients for its domain class. See the smtpd_reject_unlisted_recipient parameter description for details. This feature is available in Postfix 2.1 and later.

reject_unverified_recipient
Reject the request when mail to the RCPT TO address is known to bounce, or when the recipient address destination is not reachable. Address verification information is managed by the verify(8) server; see the ADDRESS_VERIFICATION_README file for details.
The unverified_recipient_reject_code parameter specifies the response when an address is known to bounce (default: 450, change into 550 when you are confident that it is safe to do so). Postfix replies with 450 when an address probe failed due to a temporary problem. This feature is available in Postfix 2.1 and later.

Other restrictions that are valid in this context:

- Generic restrictions that can be used in any SMTP command context, described under smtpd_client_restrictions.

- SMTP command specific restrictions described under smtpd_client_restrictions, smtpd_helo_restrictions and smtpd_sender_restrictions.

Example:

```
smtpd_recipient_restrictions = permit_mynetworks, reject_unauth_destination
```

- ## 'Greylisting' antispam module for SuSE 9.x/10.x
  - **Description:** The Greylisting AntiSpam module for Postfix will refuse all mails coming from mail servers the first time it receives them with an error code 450 that means try later. Then it will accept the mails that are resent and put their server names into a list. The next time the same server sends mail, it will be accepted the first time. So the spammers that don't use mail servers that resends mail are simply always refused. It allows also for a 'black list' of real mail servers that send spam and 'white list' of servers that will be accepted the first time.

  There are multiple implementations of this technique:

  - The **Perl based standard Greylisting system** based on perl modules and using mysql/sql-lite databases. It can be found at:
    http://projects.puremagic.com/greylisting/
  - he 'C/C++' Program using Mysql database. It can be found at:
    http://www.gasmi.net/gld.html
  - The 'C/C++' Program using DBI/Mysql databases. It can be found at:
    http://mimo.gn.apc.org/gps/
  - The 'Python' Program `tumgreyspf` using a database directly written on the local file system. It can be found at:
    http://www.tummy.com/Community/software/tumgreyspf/

  Note:I have started to describe the standard perl modules based system below but didn't finish it yet. The reason for that being that I'm not so experienced with Perl and was haing trouble finding all the Perl modules appropriate to the current perl version in SuSE 9.x/10.x.

- ## Gerylisting/SPF check based on `tumgreyspf` system.

  This system checks for bad SPF and Greylisting.
  Because of the dependance on perl modules and external databases systems of other greylisting systems I opted for the last in the above list:
  The 'Python' Program `tumgreyspf.`
  This greylisting system does also SPF checking. (address check of sender server)
  It prevents many self forged return adresses in SPAMs.
  The SPF checks the validity of the sending server which is seen in the header of the mail in 'Retrurn Path' entry.
  Note: This system uses the local file system instead of the usual MySQL database for recording the greylist. It has advantages and disadvantages.
  Advantages:

  - It doesn't rely on any external database, therefore less prone to breakdowns
  - It uses the Python interpreter instead of Perl, therefore less dependant on extra modules.

  Disadvantages:

  - The local file system is a less efficient database system as MySQL
  - The greylists must be cleaned regularly to avoid overloading the local file system.

Installation on SuSE 9.x/10.x

Note: The web site offers also an RPM for installation. The difficulty here is that the `.tar.gz` standard paths of the system are different than the ones in the RPM file. I opted for the `.tar.gz` to be more flexible in case I want to install it in a Debian system as well.

- Download the latest version of the system from:
  http://www.tummy.com/Community/software/tumgreyspf/

- Copy the tarball (`tumgreyspf-1.11 .tar.gz`) file in `/usr/local` directory.

- Unpack the file and change directory to the new unpacked directory:

```
cd /usr/local/
tar fvxz tumgreyspf-1.11.tar.gz
cd tumgreyspf-1.11
```

- Note: This system comes with a file called README. It provides useful information concerning this system. To help making sure it is adapted to the SuSE environment, I created the following instructions that I recommend to follow.

- Run the following script:(or execute all these commands one by one)

```
#!/bin/bash
TGSPROG=/usr/local/lib/tumgreyspf
TGSDATA=/var/local/lib/tumgreyspf
TGSUSER=nobody

# set up directories
cd /usr/local/tumgreyspf-1.11/
mkdir -p "$TGSPROG" "$TGSDATA"/config
mkdir "$TGSDATA"/data
cp __default__.dist "$TGSDATA"/config/__default__

# install programs
cp tumgreyspf tumgreyspf-clean tumgreyspf-configtest "$TGSPROG"
cp tumgreyspf-install tumgreyspf-stat tumgreyspfsupp.py "$TGSPROG"
cp tumgreyspf.conf "$TGSDATA"/config/
ln -s $TGSPROG/tumgreyspf-stat /usr/sbin

# change permissions and ownership
chown -R "$TGSUSER" "$TGSDATA"
chown -R root "$TGSPROG" "$TGSDATA"/config
chmod 700 "$TGSDATA"/data
chmod -R 755 "$TGSDATA"/config


# Prepare a cron job for regular daily clean-up (IMPORTANT)
echo "0 0 * * * $TGSUSER $TGSPROG/tumgreyspf-clean" \
      >/etc/cron.d/tumgreyspf
```

- Edit the file `/etc/postfix/master.cf` and add the following 2 lines:

```
tumgreyspf  unix   -    n       n       -       -       spawn
      user=nobody argv=/usr/local/lib/tumgreyspf/tumgreyspf
```

(IMPORTANT: Note that the second line doesn't start at the begining of the line)

- Edit the file `/etc/postfix/main.cf` and add the entry:

```
check_policy_service unix:private/tumgreyspf
```

right after the "reject_unauth_destination,"

Example:

```
smtpd_recipient_restrictions = \
          permit_mynetworks, \
          reject_non_fqdn_hostname, \
          reject_invalid_hostname, \
          reject_unauth_destination, \
          check_policy_service unix:private/tumgreyspf
```

WARNING: It's very important that you have "reject_unauth_destination," before the check_policy_service entry. If you don't, your system may be an open relay.

- In the same (main.cf) file add also the entry:

```
tumgreyspf_time_limit = 3600
```

(This line is undocumented, so it is recommended to enter it as it is.)

- Restart postfix with the command:

```
rcpostfix restart
```

Testing the greylisting

There is an easy way to test the greylisting using the telnet utility as follows:
Note:
In the example below, I'm initiating sending a mail from the host:
laptop.linux.site from the user billy@laptop.linux.site
to the user michel in the destination mail server vsuse93b.linux.site
The greylisting system runs in the destination mail server.
Here, what I type in the terminal is in bold, the rest are answers from the server.

```
telnet 192.168.100.40 25
Trying 192.168.100.40...
Connected to 192.168.100.40.
Escape character is '^]'.
220 vsuse93b.linux.site ESMTP Postfix
helo laptop.linux.site
250 vsuse93b.linux.site
mail from: billy@laptop.linux.site
250 Ok
rcpt to: michel@vsuse93b.linux.site
450 <michel@vsuse93b.linux.site>: Recipient address rejected:
Service unavailable, greylisted.
```

The mail was refused but the error message number 450 tells the sending server to try again later.

After 10 minutes I try again:

```
telnet 192.168.100.40 25
Trying 192.168.100.40...
Connected to 192.168.100.40.
Escape character is '^]'.
220 vsuse93b.linux.site ESMTP Postfix
helo laptop.linux.site
250 vsuse93b.linux.site
mail from: billy@laptop.linux.site
250 Ok
rcpt to: michel@vsuse93b.linux.site
250 Ok
quit
221 Bye
Connection closed by foreign host.
```

This time the mail was accepted and will always be afterwards from this server, unless it receives no mail for a certain time. Then it will be refused the first time mail again.
This time limit is set in the default configuration file explained below by the entry:
`GREYLISTEXPIREDAYS = 10.0`

Configuring the Greylisting system

This system comes with a default configuration that applies to all incoming mails and mail servers for greylisting. Extra individual configurations can also be made to override the defaults. Here are the entries and their meaning of the default configuration file locates at:

`/var/local/lib/tumgreyspf/config/__default__`

Content of default configuration file:

```
# SPFSEEDONLY=1 will only check SPF. Should not be used for decisions.
# In fact I'm not really sure what it is good for then.
SPFSEEDONLY = 0

# The time amount of time(in seconds) the mail system will be refusing a first time
# mail/mail-server before it will accept any mail from this server forever afterwards.
# In this case a server can retry sending the mail 10 minutes later and it will be accepted.
GREYLISTTIME = 600

# what checks will be performed on all mails. Only the listed checks will be performed.
#greylist   Performs a check against the greylist
#spf        Performs an SPF check in the mail header
#blackhole  Performs a Blacklist check to refuse a specific email based on the IP
#           or the sender's address.
CHECKERS = greylist,spf,blackhole

# Which configurations are taken for accounts when checking
OTHERCONFIGS = client_address,envelope_sender,envelope_recipient

#  The number of days after which, if no messages have come in from a server
#  we will drop the greylist entry. That means blocking again the first attempt to send mail
#  from this server. This value is used by "tumgreyspf-clean" program normally run by
#  a cron job.
```

```
GREYLISTEXPIREDAYS = 10.0
```

Creating while lists

(for servers that shouldn't be refused first time mail)

We can 'whitelist' 4 types of information:

- Single IP
- Full subnet (eg. 192.168.100.0/24)
- Recipient user address
  (contained in the email header 'envelope' not the 'To: ...' in the message'
- Sender user address
  (contained in the email header 'envelope' not the 'From: ...' in the message'

Whitelisting an IP of a remote mail server.

If a server doesn't respond well to the 'Resend Later' error message `450` and doesn't resend later, then we need to enter its IP into a while list that will let it send emails without firs time refusal. White listing is done by creating a configuration file in a specific directory. Here is an example:

If we want to always allow mail from the host with IP `213.56.156.23` but still check its SPF(`CHECKERS=spf`) we would create the file:

`/var/local/lib/tumgreyspf/config/client_address/213/56/156/`**`23`**

the file named ' **23** ' would contain the following lines:

```
SPFSEEDONLY=0
GREYLISTTIME=300
CHECKERS=spf
OTHERCONFIGS=
```

Now that is a bit of work to do for each IP we want to 'whitelist'. So I've created the following small bash script that does the job.
Syntax:

`whitelist-ip` *`IPNumber`*

eg.

`whitelist-ip 213.56.156.23`

```
#!/bin/bash
# Creates  a whitelist of an IP for tumgreyspf system
# Make sure that we have one parameter

#Setting some variables
whitelistdir="/var/local/lib/tumgreyspf/config/client_address"
IP=$1
# Make sure we have one and only one parameter as the IP
if [ "$#" -ne 1 ]; then
      echo "ERROR: Wrong number of parameters"
      echo "Syntax:  whitelist-ip   IPNumber"
      exit 1
fi
# Make sure that the IP given is a valid IP
if !(echo $IP | egrep "^([0-9]{1,3}\.){3}[0-9]{1,3}$" &>/dev/null) ; then
      echo "ERROR: Bad IP Syntax"
      exit 2
fi

#-----------------------------------------------------------------
# Verify validity if all numbers in IP (0-255)
IFS="."
len=0
```

```
for num in $IP ; do
    let len++
    # Do not accept more than 4 numbers
    if [ "$len" -gt 4 -a "$num" != "" ] ; then
          echo "ERROR: NO proper IP given."
          exit 3
    # Do not accept numbers higher than 255
    elif [ "$num" -gt 255 ] ; then
          echo "ERROR: Wrong values in IP."
          exit 4
    # Do not accept empty fields eg. 192..168.30
    elif [ "$num" = "" ] ; then
          echo "ERROR: Wrong format IP."
          exit 5
    fi
done
unset IFS
# Extract the IP part that will be used as a dircectory name
dirpart=$(echo $IP | cut -d. -f1,2,3 | tr "." "/")
mkdir -p $whitelistdir/$dirpart 2>/dev/null
configfilename=$(echo $IP | cut -d. -f4)

# Now create the configuration file(whitelisting) for this IP
echo "PFSEEDONLY=0"     >  $whitelistdir/$dirpart/$configfilename
echo "GREYLISTTIME=300" >> $whitelistdir/$dirpart/$configfilename
echo "CHECKERS=spf"     >> $whitelistdir/$dirpart/$configfilename
echo "OTHERCONFIGS="    >> $whitelistdir/$dirpart/$configfilename
```

Whitelisting an subnet of a remote mail server.

A full subnet can be 'whitelisted' by creating a `__default__` configuration file with the same content as the one for 'whitelisting' an IP in following manner:

Example: If we want to 'whitelist' all hosts from the local subnet `192.168.100.0/24` then we would create the following `_default_` file:
`/var/local/lib/tumgreyspf/config/client_address/192/168/100/__default__`

In this case the SPF check does not need to be performed since it is most likely our local network. (CHECKERS=)
This `__default__` file would contain:

```
SPFSEEDONLY=0
GREYLISTTIME=300
CHECKERS=
OTHERCONFIGS=
```

I've created the following small bash script that does the job.
Syntax:

`whitelist-net` *`PartialIPNumber`*

eg.

`whitelist-net 192.168.100`

```
#!/bin/bash
# Creates  a whitelist of all hosts of a subnet for tumgreyspf system
# Make sure that we have one parameter

#Setting some variables

IP=$1
whitelistdir="/var/local/lib/tumgreyspf/config/client_address"
# Make sure we have one and only one parameter as the Partial IP
if [ "$#" -ne 1 ]; then
      echo "ERROR: Wrong number of parameters"
      echo "Syntax:  whitelist-net  PartialIPNumber"
      exit 1
```

```
fi
# Make sure that the partial IP given is valid
if !(echo $IP | egrep "^([0-9]{1,3}\.){2}[0-9]{1,3}$" &>/dev/null) ; then
      echo "ERROR: Bad partial IP Syntax"
      exit 2
fi
# Verify validity if all numbers in IP (0-255)
IFS="."
len=0
for num in $IP ; do
    let len++
    # Do not accept more than 3 numbers
    if [ "$len" -gt 3 -a "$num" != "" ] ; then
          echo "ERROR: NO proper IP given."
          exit 3
    # Do not accept numbers higher than 255
    elif [ "$num" -gt 255 ] ; then
          echo "ERROR: Wrong values in IP."
          exit 4
    # Do not accept empty fields eg. 192..168
    elif [ "$num" = "" ] ; then
          echo "ERROR: Wrong format in IP."
          exit 5
    fi
done
unset IFS
# Extract the IP part that will be used as a directory name
dirpart=$(echo $IP | cut -d. -f1,2,3 | tr "." "/")
mkdir -p $whitelistdir/$dirpart 2>/dev/null

# Now create the configuration file(whitelisting) for this Network
echo "PFSEEDONLY=0"     >  $whitelistdir/$dirpart/__default__
echo "GREYLISTTIME=300" >> $whitelistdir/$dirpart/__default__
echo "CHECKERS="        >> $whitelistdir/$dirpart/__default__
echo "OTHERCONFIGS="    >> $whitelistdir/$dirpart/__default__
```

Whitelisting a recipient's address.

If we want to always allow all incoming mails for a local user from the first time on, then we would create a configuration file called after the user containing the same as for an IP whitelisting. Example: Always allowing all incoming emails for address: `martin@mydomain.com` then we would create the file:

`/var/local/lib/tumgreyspf/config/envelope_recipient/mydomain.com/martin`

with the content:

```
SPFSEEDONLY=0
GREYLISTTIME=300
CHECKERS=spf
OTHERCONFIGS=
```

I've created the following small bash script that does the job.

Syntax:

`whitelist-recipient` *`RecipientAddress`*

eg.

`whitelist-recipient martin@mydomain.com`

Whitelisting a recipient's address (whitelist-recipient)

```
#!/bin/bash
# Creates  a whitelist of a recipient's adddress for tumgreyspf system
# Make sure that we have one parameter

#Setting some variables
addr=$1
whitelistdir="/var/local/lib/tumgreyspf/config/envelope_recipient"


# Make sure we have one and only one parameter as the recipient's address
if [ "$#" -ne 1 ]; then
      echo "ERROR: Wrong number of parameters"
      echo "Syntax:  whitelist-recipient  RecipientAddress"
      exit 1
fi
# Make sure that the recipient address is a valid email address format
if !(echo $addr | egrep "^.+@.+\..+$" &>/dev/null) ; then
      echo "ERROR: Bad partial email address Syntax"
      exit 2
fi

#-----------------------------------------------------------------
# Extract the host part that will be used as a directory name
dirpart=$(echo $addr | cut -d@ -f2)
username=$(echo $addr | cut -d@ -f1)
mkdir -p $whitelistdir/$dirpart 2>/dev/null

# Now create the configuration file(whitelisting) for this Network
echo "PFSEEDONLY=0"     >  $whitelistdir/$dirpart/$username
echo "GREYLISTTIME=300" >> $whitelistdir/$dirpart/$username
echo "CHECKERS=spf"     >> $whitelistdir/$dirpart/$username
echo "OTHERCONFIGS="    >> $whitelistdir/$dirpart/$username
```

Whitelisting a sender's address.

'Whitelisting' a sender's address is the same principle as for a recipient's address except that the subdirectory name is `envelope_recipient` instead of `envelope_sender`.

Example: Always allowing all incoming emails coming from address: `eveline@jolie.com` then we would create the file:

```
/var/local/lib/tumgreyspf/config/envelope_sender/jolie.com/eveline
```

with the content:

```
SPFSEEDONLY=0
GREYLISTTIME=300
CHECKERS=spf
OTHERCONFIGS=
```

I've created the following small bash script that does the job.

Syntax: `whitelist-sender` *`SendertAddress`*

eg. `whitelist-sender eveline@jolie.com`

Whitelisting a sender's address (whitelist-sender)

```
#!/bin/bash
# Creates  a whitelist of a sender's adddress for tumgreyspf system
# Make sure that we have one parameter

#Setting some variables

addr=$1
whitelistdir="/var/local/lib/tumgreyspf/config/envelope_sender"


# Make sure we have one and only one parameter as the sender's address
if [ "$#" -ne 1 ]; then
      echo "ERROR: Wrong number of parameters"
      echo "Syntax:  whitelist-sender     SenderMailAddress"
      exit 1
fi
# Make sure that the sender address is a valid email address format
if !(echo $addr | egrep "^.+@.+\..+$" &>/dev/null) ; then
      echo "ERROR: Bad partial email address Syntax"
      exit 2
fi

#-----------------------------------------------------------------
# Extract the host part that will be used as a directory name
dirpart=$(echo $addr | cut -d@ -f2)

# create the directory
mkdir -p $whitelistdir/$dirpart 2>/dev/null

# Extract the username from the email address
username=$(echo $addr | cut -d@ -f1)

# Now create the configuration file(whitelisting) for this user
echo "PFSEEDONLY=0"     >  $whitelistdir/$dirpart/$username
echo "GREYLISTTIME=300" >> $whitelistdir/$dirpart/$username
echo "CHECKERS=spf"     >> $whitelistdir/$dirpart/$username
echo "OTHERCONFIGS="    >> $whitelistdir/$dirpart/$username
```

Blacklisting IP addresses.

To allow for 'Blackhole' checking, the word '`blackhole`' MUST be in the list of checks in the main `__default__` configuration file.

```
CHECKERS=spf, blackhole
```

eg. Blacklisting the IP address: 243.57.139.30 and 210.57.21.37

Create 2 empty files called:

```
/var/lib/tumgreyspf/blackhole/ips/243.57.139.30
/var/lib/tumgreyspf/blackhole/ips/210.57.21.37
```

Blacklisting sender addresses:

To allow for 'Blackhole' checking, the word '`blackhole`' MUST be in the list of checks in the main `__default__` configuration file.

```
CHECKERS=spf, blackhole
```

eg. Blacklisting the sender address: `malware@blackmec.sk` and `joe@party.com`

```
/var/lib/tumgreyspf/blackhole/addresses/malware@blackmec.sk
/var/lib/tumgreyspf/blackhole/addresses/joe@party.com
```

Getting a Greylisting status

There is a program that is provided with this system that displays the status of the greylisting. The program is called:

```
/usr/sbin/tumgreyspf-stat
```

This is a symbolic link to `/usr/local/lib/tumgreyspf/tumgreyspf-stat.`

The format of the result of status is on e entry per line and each line is as follows:

eg.

```
IP=84.23.136.61 SENDER=ddzm@rhi.com RECIPIENT=prod@bild.de STARTS=-30 LAST=569 EXPIRESIN=-864000
(Blocked,Pending)
-------------- ---------------------- ---------------------- ------------ --------- ----------------- ---------------
```

A B C D E F G

A = IP of server sending the mail.

B = Address of Sender

C = Address of local recipient

D = Pending time (in seconds) left before the mail could be accepted (Blocking period)

E = Elapsed Time (in seconds) since the last attempt to send the mail from the sending remote server.

F = Period of Time (in seconds) this email will be registered. If no enails are received from this server inside this period of time then the IP is cleaned-up from the system. Any new mail afterwards from this server will be rejected the first time and after the pending time is over the emails will then be accepted again.

G = Status of the registration:

(Blocked,Pending) = Email has been rejected and is pending its acceptance time

(Blocked) = This email can now be accepted if resent from server but has not been resent from the server yet.

Nothing = All emails sent from this server to this recipient will from now on be accepted.

Perl based standard Greylisting system (not finished yet)

More information on this systemcan be found at:

- **Installation for working with MySQL:**
  Get from `http://rpm.pbone.net` and install the latest RPM versions of:
  `sqlgrey`
  `rpm-helper` ----> Just ignore the dependencies with SuSE 9.3
  They are satisfied through other packages.
  `IO::Multiplex` Perl Module
- Install the following packages from the SuSE 9.3 CD/DVD:
  - `mysql`
  - `mysql-client`
  - `perl-DBD-mysql` (Perl module)
- Create a group called `sqlgrey:` Command : `groupadd sqlgrey`
- Create a user called `sqlgrey`. Command: `useradd -g sqlgrey sqlgrey`
- Change the database type in `/etc/sqlgrey/sqlgrey.conf:`

```
db_type = mysql
db_name = sqlgrey
db_host = localhost
db_user = sqlgrey
db_pass = spaces_are_not_supported
db_cleandelay = 1800
```

- Configure the rest of `/etc/sqlgrey/sqlgrey.conf` as desired.
  eg. email notifications of server status.
  `admin_mail = michel@linuxint.com`
- Create a `sqlgrey` database in MySQL:
  `mysql -u root -p` (Then give the mysql root password)

```
> CREATE DATABASE sqlgrey;
> GRANT ALL ON sqlgrey.* TO sqlgrey@localhost;
> quit
```

- **In POSTFIX**
  Add `check_policy_service` after `reject_unauth_destination` in `/etc/postfix/main.cf`

eg.

```
smtpd_recipient_restrictions =.....
     reject_unauth_destination,
     check_policy_service inet:127.0.0.1:2501
```

This assumes `sqlgrey` will listen on the TCP 2501 port (default) and is on the same host.

- **STARTING SQLGREY**
  Note: `sqlgrey` version 1.6.0 installs an init script in `/etc/rc.d/init.d.`
  It doesn't work in SuSE. You need to use the script on the next page
  and save it in `/etc/init.d/sqlgrey`
  To make sure it starts at boot time: `insserv sqlgrey`
  `sqlgrey` should be started via this init script: `/etc/init.d/sqlgrey`
  It will send its logs as mail log.(`tail -f /var/log/mail`)

DNS-Hilfprogramme

`host [-v] Rechnername`	versucht der Rechnernamen aufzulösen. -v = verbose, die Ausgabe ist dann ähnlich wie mit `dig`.
`host Rechnername DNS-Server`	benutzt den angegeben DNS-Server für die Auflösung
`host IP-Adresse`	versucht die IP-Adresse aufzulösen.
`host -l Domäne`	zeigt alle Rechner einer DNS-Domäne.
`host -t mx Domäne`	zeigt der Mail-Exchange-Server einer Domäne.
`dig [@server] name [type]`	`dig` wie `host` erlaubt einen Rechnernamen aufzulösen, aber gibt mehr Informationen. (type = any, a, mx, ns usw.)
`dig sun.linux.local`	versucht `sun.linux.local` aufzulösen
`dig @dozlinux sun`	versucht `sun` vom DNS-Server `dozlinux` aufzulösen.
`dig linux.local any`	zeigt die ganze Domäne `linux.local` an.
`dig -x IP-Adresse`	versucht eine IP-Adresse aufzulösen.

• Postfix basic exercises

1) access

-edit `/etc/postfix/access` file and enter

```
michel@bts02doz.linux.local    REJECT
```

- run the commands

```
postmap /etc/postfix/access
rcpostfix restart
```

- run `tail -f /var/log/mail` in a terminal on the server
- send a mail from `michel@bts02doz.linux.local` to root at the server
- see the mail rejected

2) alias

- make sure there is `admin` user in the local server
- modify the `/etc/aliases` to include

```
mailuser1:     root
mailuser2:          admin
```

- run the commands:

```
newaliases
rcpostfix restart
```

- `mail  mailuser1`
- `mail  mailuser2`
- `su -`

`mail` (mail to mailuser1 should be there)

`su - admin`

`mail` (mail to mailuser2 should be there)

3)canonical

- edit the file `/etc/postfix/canonical` and enter:

```
root.admin     root
```

- run the commands:

```
postmap /etc/postfix/canonical
rcpostfix restart
```

- send a mail to `root.admin@`*`mailserver`*`.linux.local`
- see the mail arriving on the server in `root` user mailbox

4)relocated

-edit the file `/etc/postfix/relocated` and enter:

```
user1     user1@newcompany.de Please make note of it
```

- run the commands:

```
postmap /etc/postfix/relocated
rcpostfix restart
```

- send a mail to `user1@`*`mailserver`*`.linux.local`
- see the mail being bounced and back in the client sender mailbox

5)virtual

- Make sure that the MX record in DNS is set to:

```
special.linux.local     IN MX     mailserver.linux.local.
special.linux.local     IN CNAME  mailserver.linux.local.
mailserver.linux.local. IN A      192.168.xxx.yyy
```

- Edit the file /etc/postfix/virtual on mailserver and enter:

```
special.linux.local virtual
myuser@special.linux.local    user1
```

- Run the commands:

```
postmap /etc/postfix/virtual
rcpostfix restart
```

- Send a mail from client to myuser@special.linux.local

- Check the mail of user1 on mailserver. The mail should be there.

- **Tests of 3 computers as:**
  - client(win/linux) (pop3 account in the local mail server)
  - local mail server (fetchmail the ISP through pop3, plus pop3/IMAP server)
  - ISP/Mail server (pop3 server)

- **Example of Mail header including MIME**

```
sendmail michel.dozlinux.local
Subject: hallo in html
Mime-Version: 1.0
Content-type: text/html
<Body><H1><Font color=red>
     hallo world
</Font></H1></body>
```

Introduction

Although the initial Postfix release has no address rewriting language, it can do quite a bit of address manipulation via table lookup. While a message flows through the Postfix system, its addresses are mangled in the order described in this document.
Unless indicated otherwise, all parameters described here are in the `main.cf` file. If you change parameters of a running Postfix system, don't forget to issue a `postfix reload` command.

All mail:

- Rewrite addresses to standard form
- Canonical address mapping
- Address masquerading
- Virtual address mapping
- Mail transport switch
- Relocated users table

Local delivery:

- Alias database
- Per-user .forward files
- Non-existent users

Rewrite addresses to standard form

Before the cleanup daemon runs an address through any lookup table, it first rewrites the address to the standard *`user@fully.qualified.domain`* form, by sending the address to the trivial-rewrite daemon. The purpose of rewriting to standard form is to reduce the number of entries needed in lookup tables. The Postfix trivial-rewrite program implements the following hard-coded address manipulations:

Rewrite *`@hosta,@hostb:user@site`* to *`user@site`*
: The source route feature has been deprecated. Postfix has no ability to handle such addresses, other than to strip off the source route.

Rewrite *`site!user`* to *`user@site`*
: This feature is controlled by the boolean `swap_bangpath` parameter (default: `yes`). The purpose is to rewrite **UUCP**-style addresses to domain style. This is useful only when you receive mail via **UUCP**, but it probably does not hurt otherwise.

Rewrite *`user%domain`* to *`user@domain`*
: This feature is controlled by the boolean `allow_percent_hack` parameter (default: `yes`). Typically, this is used in order to deal with monstrosities such as *`user%domain@otherdomain`*.

Rewrite *`user`* to *`user@$myorigin`*
: This feature is controlled by the boolean `append_at_myorigin` parameter (default: `yes`). The purpose is to get consistent treatment of *user* on every machine in `$myorigin`.

  You probably should never turn off this feature, because a lot of Postfix components expect that all addresses have the form *`user@domain`*.

  If your machine is not the main machine for **$myorigin** and you wish to have some users delivered locally without going via that main machine, make an entry in the

virtual table that redirects *user@$myorigin* to *user@$myhostname*.

Rewrite *user@host* to *user@host.$mydomain*
This feature is controlled by the boolean `append_dot_mydomain` parameter (default: `yes`). The purpose is to get consistent treatment of different forms of the same hostname.

Some will argue that rewriting *host* to *host.$mydomain* is bad. That is why it can be turned off. Others like the convenience of having the local domain appended automatically.

Rewrite *user@site.* to *user@site* (without the trailing dot).

Canonical address mapping

Before the cleanup daemon stores inbound mail into the **incoming** queue, it uses the canonical table to rewrite all addresses in message envelopes and in message headers, local or remote. The mapping is useful to replace login names by *Firstname.Lastname* style addresses, or to clean up invalid domains in mail addresses produced by legacy mail systems.

Canonical mapping is disabled by default. To enable, edit the `canonical_maps` parameter in the **main.cf** file and specify one or more lookup tables, separated by whitespace or commas. For example:

```
canonical_maps = hash:/etc/postfix/canonical
```

In addition to the canonical maps which are applied to both sender and recipient addresses, you can specify canonical maps that are applied only to sender addresses or to recipient addresses. For example:

```
sender_canonical_maps = hash:/etc/postfix/sender_canonical
recipient_canonical_maps = hash:/etc/postfix/recipient_canonical
```

The sender and recipient canonical maps are applied before the common canonical maps. Sender-specific rewriting is useful when you want to rewrite ugly sender addresses to pretty ones, and still want to be able to send mail to the those ugly address without creating a mailer loop.

Address masquerading

Address masquerading is a method to hide all hosts inside a domain behind their mail gateway, and to make it appear as if the mail comes from the gateway itself, instead of from individual machines. Address masquerading is disabled by default. To enable, edit the `masquerade_domains` parameter in the **`main.cf`** file and specify one or more domain names separated by whitespace or commas. The list is processed left to right, and processing stops at the first match. Thus,

```
masquerade_domains = foo.example.com example.com
```

strips `any.thing.foo.example.com` to `foo.example.com`, but strips `any.thing.else.example.com` to `example.com`.

A domain name prefixed with `!` means do not masquerade this domain or its subdomains. Thus,

```
masquerade_domains = !foo.example.com example.com
```

does not change `any.thing.foo.example.com` and `foo.example.com`, but strips `any.thing.else.example.com` to `example.com`.

The `masquerade_exceptions` configuration parameter specifies what user names should not be subjected to address masquerading. Specify one or more user names separated by whitespace or commas. For example,

```
masquerade_exceptions = root
```

By default, Postfix makes no exceptions.
Subtle point: by default, address masquerading is applied only to message headers and to envelope sender addresses, but not to envelope recipients. This allows you to use address masquerading on a mail gateway machine, while still being able to forward mail from outside to users on individual machines. In order to subject envelope recipient addresses to masquerading, too, specify (only available with Postfix versions after 20010802):

```
masquerade_classes = envelope_sender, envelope_recipient,
                     header_sender, header_recipient
```

If you do this, Postfix will no longer be able to send mail to individual machines.

Virtual address aliasing

After applying the canonical and masquerade mappings, the cleanup daemon uses the virtual alias table to redirect mail for all recipients, local or remote. The mapping affects only envelope recipients; it has no effect on message headers or envelope senders. Virtual alias lookups are useful to redirect mail for simulated virtual domains to real user mailboxes, and to redirect mail for domains that no longer exist. Virtual alias lookups can also be used to transform *`Firstname.Lastname`* back into UNIX login names, although it seems that local aliases are a more appropriate vehicle.
Virtual aliasing is disabled by default. To enable, edit the `virtual_alias_maps` parameter in the **`main.cf`** file and specify one or more lookup tables, separated by whitespace or commas. For example:

```
virtual_alias_maps = hash:/etc/postfix/virtual
```

Addresses found in virtual alias maps are subjected to another iteration of virtual aliasing, but are not subjected to canonical mapping, in order to avoid loops.

Mail transport switch

Once the address rewriting and resolving daemon has established the destination of a message, it determines the default delivery method for that destination. Postfix distinguishes four major address classes, each with its own default delivery method.

Destination matches	Default delivery agent	Controlling parameter
`$mydestination` or `$inet_interfaces`	local	**`$local_transport`**
`$virtual_mailbox_domains`	virtual	**`$virtual_transport`**
`$relay_domains`	relay (clone of smtp)	**`$relay_transport`**
none	smtp	**`$default_transport`**

The optional transport table overrides the default message delivery method (this table is used by the address rewriting and resolving daemon). The transport table can be used to send mail to specific sites via **UUCP**, or to send mail to a really broken mail system that can handle only one SMTP connection at a time (yes, such systems exist and people used to pay real money for them).

Transport table lookups are disabled by default. To enable, edit the `transport_maps` parameter in the **`main.cf`** file and specify one or more lookup tables, separated by whitespace or commas. For example:

```
transport_maps = hash:/etc/postfix/transport
```

Relocated users table

Next, the address rewriting and resolving daemon runs each recipient name through the relocated database. This table provides information on how to reach users that no longer have an account, or what to do with mail for entire domains that no longer exist. When mail is sent to an address that is listed in this table, the message is bounced with an informative message.

Lookups of relocated users are disabled by default. To enable, edit the `relocated_maps` parameter in the **`main.cf`** file and specify one or more lookup tables, separated by whitespace or commas. For example:

```
relocated_maps = hash:/etc/postfix/relocated
```

Alias database

When mail is to be delivered locally, the local delivery agent runs each local recipient name through the aliases database. The mapping does not affect addresses in message headers. Local aliases are typically used to implement distribution lists, or to direct mail for standard aliases such as **`postmaster`** to real people. The table can also be used to map *`Firstname.Lastname`* addresses to login names.

Alias lookups are enabled by default. The default configuration depends on the system environment, but it is typically one of the following:

```
alias_maps = hash:/etc/aliases
alias_maps = dbm:/etc/aliases, nis:mail.aliases
```

The path to the alias database file is controlled via the `alias_database` configuration parameter. The value is system dependent. Usually it is one of the following:

```
alias_database = hash:/etc/aliases (4.4BSD, LINUX)
alias_database = dbm:/etc/aliases (4.3BSD, SYSV<4)
alias_database = dbm:/etc/mail/aliases (SYSV4)
```

For security reasons, deliveries to command and file destinations are performed with the rights of the alias database owner. A default userid, **default_privs**, is used for deliveries to commands/files in *root*-owned aliases.

Per-user `.forward` files

Users can control their own mail delivery by specifying destinations in a file called **`.forward`** in their home directories. The syntax of these files is the same as with system aliases, except that the lookup key and colon are not present.

Non-existent users

When the local delivery agent finds that a message recipient does not exist, the message is normally bounced to the sender ("user unknown"). Sometimes it is desirable to forward mail for non-existing recipients to another machine. For this purpose you can specify an alternative destination with the `luser_relay` configuration parameter.
Alternatively, mail for non-existent recipients can be delegated to an entirely different message transport, as specified with the `fallback_transport` configuration parameter. For details, see the local delivery agent.

Note: if you use the `luser_relay` feature in order to receive mail for non-UNIX accounts, then you must specify:

```
local_recipient_maps =
```

(i.e. empty) in the **`main.cf`** file, otherwise the Postfix SMTP server will reject mail for non-UNIX accounts with "User unknown in local recipient table".

`luser_relay` can specify one address. It is subjected to *$name* expansions.
The most useful examples are:

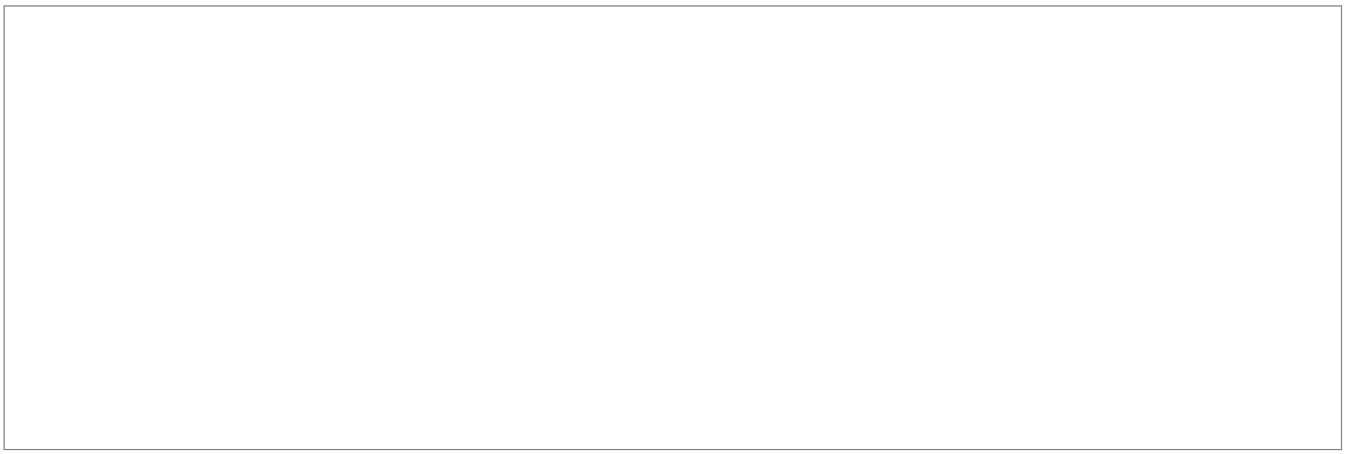
`$user@other.host`
: The bare username, without address extension, is prepended to *`@other.host`*. For example, mail for *`username+foo`* is sent to *`username@other.host`*.

`$mailbox@other.host`
: The entire original recipient localpart, including address extension, is prepended to *`@other.host`*. For example, mail for *`username+foo`* is sent to *`username+foo@other.host`*.

`sysadmin+$user`
: The bare username, without address extension, is appended to *sysadmin*. For example, mail for *`username+foo`* is sent to *`sysadmin+username`*.

`sysadmin+$mailbox`
: The entire original recipient localpart, including address extension, is appended to *`sysadmin`*. For example, mail for *`username+foo`* is sent to *`sysadmin+username+foo`*.

Postfix - the Big Picture



The figure shows the main Postfix system components, and the main information flows between them. Postfix system components are introduced in the Postfix anatomy documentation.

- Yellow ellipsoids are mail programs.
- Yellow boxes are mail queues or files.
- Blue boxes are lookup tables.
- Programs in the large box run under control by the Postfix resident master daemon.
- Data in the large box is property of the Postfix mail system.

In order to keep the big picture readable the following elements were omitted:

- The Postfix command-line utilities.
- The Postfix resident master daemon.
- The DNS lookups by the SMTP server and client daemons
- The bounce or defer daemon and the flow of bounced mail.
- The address rewriting and resolving requests by the SMTP server and by the local delivery agent.
- The flow of mail forwarded by the local delivery agent.
- The flow of postmaster notices for protocol errors, policy violations, etc.
- Triggers to alert the pickup daemon and queue manager that new mail has arrived in the **maildrop** and **incoming** queues, respectively.

Receiving Mail

When a message enters the Postfix mail system, the first stop on the inside is the incoming queue. The figure below shows the main components that are involved with new mail. For an explanation of the symbols used, click on the icon in the upper left-hand corner of this page.

- Mail is posted locally. The Postfix sendmail program invokes the privileged postdrop program which deposits the message into the **maildrop** directory, where the message is picked up by the pickup daemon. This daemon does some sanity checks, in order to protect the rest of the Postfix system.
- Mail comes in via the network. The Postfix SMTP server receives the message and does some sanity checks, in order to protect the rest of the Postfix system. The SMTP server can be configured to implement UCE controls on the basis of local or network-based black lists, DNS lookups, and other client request information.
- Mail is generated internally by the Postfix system itself, in order to return undeliverable mail to the sender. The bounce or defer daemon brings the bad news.
- Mail is forwarded by the local delivery agent, either via an entry in the system-wide alias database, or via an entry in a per-user .forward file. This is indicated with the unlabeled arrow.
- Mail is generated internally by the Postfix system itself, in order to notify the postmaster of a problem (this path is also indicated with the unlabeled arrow). The Postfix system can be configured to notify the postmaster of SMTP protocol problems, UCE policy violations, and so on.
- The cleanup daemon implements the final processing stage for new mail. It adds missing **From:** and other message headers, arranges for address rewriting to the standard *user@fully.qualified.domain* form, and optionally extracts recipient addresses from message headers. The **cleanup** daemon inserts the result as a single queue file into the **incoming** queue, and notifies the queue manager of the arrival of new mail. The **cleanup** daemon can be configured to transform addresses on the basis of canonical and virtual table lookups.
- On request by the **cleanup** daemon, the trivial-rewrite daemon rewrites addresses to the standard *user@fully.qualified.domain* form. The initial Postfix version does not implement a rewriting language. Implementing one would take a lot of effort, and most sites do not need it. Instead, Postfix makes extensive use of table lookup.

SMTPD(8) SMTPD(8)

NAME

smtpd - Postfix SMTP server

SYNOPSIS

smtpd [generic Postfix daemon options]

DESCRIPTION

The SMTP server accepts network connection requests and performs zero or more SMTP transactions per connection. Each received message is piped through the **cleanup**(8) daemon, and is placed into the **incoming** queue as one single queue file. For this mode of operation, the program expects to be run from the **master**(8) process manager.

Alternatively, the SMTP server takes an established connection on standard input and deposits messages directly into the **maildrop** queue. In this so-called stand-alone mode, the SMTP server can accept mail even while the mail system is not running.

The SMTP server implements a variety of policies for connection requests, and for parameters given to **HELO, ETRN, MAIL FROM, VRFY** and **RCPT TO** commands. They are detailed below and in the **main.cf** configuration file.

SECURITY

The SMTP server is moderately security-sensitive. It talks to SMTP clients and to DNS servers on the network. The SMTP server can be run chrooted at fixed low privilege.

STANDARDS

RFC 821 (SMTP protocol)
RFC 1123 (Host requirements)
RFC 1652 (8bit-MIME transport)
RFC 1869 (SMTP service extensions)
RFC 1870 (Message Size Declaration)
RFC 1985 (ETRN command)
RFC 2554 (AUTH command)
RFC 2821 (SMTP protocol)
RFC 2920 (SMTP Pipelining)

DIAGNOSTICS

Problems and transactions are logged to **syslogd**(8).

Depending on the setting of the **notify_classes** parameter, the postmaster is notified of bounces, protocol problems, policy violations, and of other trouble.

CONFIGURATION PARAMETERS

The following **main.cf** parameters are especially relevant to this program. See the Postfix **main.cf** file for syntax details and for default values. Use the **postfix reload** command after a configuration change.

Compatibility controls

strict_rfc821_envelopes
Disallow non-RFC 821 style addresses in SMTP commands. For example, the RFC822-style address forms with comments that Sendmail allows.

broken_sasl_auth_clients
Support older Microsoft clients that mis-implement the AUTH protocol, and that expect an EHLO response of "250 AUTH=list" instead of "250 AUTH list".

smtpd_noop_commands
List of commands that are treated as NOOP (no operation) commands, without any parameter syntax checking and without any state change. This list overrides built-in command definitions.

Content inspection controls

content_filter
The name of a mail delivery transport that filters mail and that either bounces mail or re-injects the result back into Postfix. This parameter uses the same syntax as the right-hand side of a Postfix transport table.

Authentication controls

enable_sasl_authentication
Enable per-session authentication as per RFC 2554 (SASL). This functionality is available only when explicitly selected at program build time and explicitly enabled at runtime.

smtpd_sasl_local_domain
The name of the local authentication realm.

smtpd_sasl_security_options
Zero or more of the following.

noplaintext
Disallow authentication methods that use plaintext passwords.

noactive
Disallow authentication methods that are vulnerable to non-dictionary active attacks.

nodictionary
Disallow authentication methods that are vulnerable to passive dictionary attack.

noanonymous
Disallow anonymous logins.

smtpd_sender_login_maps
Maps that specify the SASL login name that owns a MAIL FROM sender address. Used by the **reject_sender_login_mismatch** sender anti-spoofing restriction.

Miscellaneous

always_bcc
Address to send a copy of each message that enters the system.

authorized_verp_clients
Hostnames, domain names and/or addresses of clients that are authorized to use the XVERP extension.

debug_peer_level
Increment in verbose logging level when a remote host matches a pattern in the **debug_peer_list** parameter.

debug_peer_list
List of domain or network patterns. When a remote host matches a pattern, increase the verbose logging level by the amount specified in the **debug_peer_level** parameter.

default_verp_delimiters
The default VERP delimiter characters that are used when the XVERP command is specified without explicit delimiters.

error_notice_recipient
Recipient of protocol/policy/resource/software error notices.

hopcount_limit
Limit the number of **Received:** message headers.

notify_classes
List of error classes. Of special interest are:

policy When a client violates any policy, mail a transcript of the entire SMTP session to the postmaster.

protocol
When a client violates the SMTP protocol or issues an unimplemented command, mail a transcript of the entire SMTP session to the postmaster.

smtpd_banner
Text that follows the **220** status code in the SMTP greeting banner.

smtpd_expansion_filter
Controls what characters are allowed in $name expansion of rbl template responses and other text.

smtpd_recipient_limit
Restrict the number of recipients that the SMTP server accepts per message delivery.

smtpd_timeout
Limit the time to send a server response and to receive a client request.

soft_bounce
Change hard (5xx) reject responses into soft (4xx) reject responses. This can be useful for testing purposes.

verp_delimiter_filter
The characters that Postfix accepts as VERP delimiter characters.

Known_versus_unknown_recipients

show_user_unknown_table_name
Whether or not to reveal the table name in the "User unknown" responses. The extra detail makes trouble shooting easier but also reveals information that is nobody elses business.

unknown_local_recipient_reject_code
The response code when a client specifies a recipient whose domain matches **$mydestination** or **$inet_interfaces**, while **$local_recipient_maps** is non-empty and does not list the recipient address

```
                or address local-part.
```

unknown_relay_recipient_reject_code
The response code when a client specifies a recipient whose domain matches **$relay_domains**, while **$relay_recipient_maps** is non-empty and does not list the recipient address.

unknown_virtual_alias_reject_code
The response code when a client specifies a recipient whose domain matches **$virtual_alias_domains**, while the recipient is not listed in **$virtual_alias_maps**.

unknown_virtual_mailbox_reject_code
The response code when a client specifies a recipient whose domain matches **$virtual_mailbox_domains**, while the recipient is not listed in **$virtual_mailbox_maps**.

Resource controls

line_length_limit
Limit the amount of memory in bytes used for the handling of partial input lines.

message_size_limit
Limit the total size in bytes of a message, including on-disk storage for envelope information.

queue_minfree
Minimal amount of free space in bytes in the queue file system for the SMTP server to accept any mail at all.

smtpd_history_flush_threshold
Flush the command history to postmaster after receipt of RSET etc. only if the number of history lines exceeds the given threshold.

Tarpitting

smtpd_error_sleep_time
Time to wait in seconds before sending a 4xx or 5xx server error response.

smtpd_soft_error_limit
When an SMTP client has made this number of errors, wait *error_count* seconds before responding to any client request.

smtpd_hard_error_limit
Disconnect after a client has made this number of errors.

smtpd_junk_command_limit
Limit the number of times a client can issue a junk command such as NOOP, VRFY, ETRN or RSET in one SMTP session before it is penalized with tarpit delays.

UCE control restrictions

parent_domain_matches_subdomains
List of Postfix features that use *domain.tld* patterns to match *sub.domain.tld* (as opposed to requiring *.domain.tld* patterns).

smtpd_client_restrictions
Restrict what clients may connect to this mail system.

smtpd_helo_required
Require that clients introduce themselves at the beginning of an SMTP session.

smtpd_helo_restrictions
Restrict what client hostnames are allowed in **HELO** and **EHLO** commands.

smtpd_sender_restrictions
Restrict what sender addresses are allowed in **MAIL FROM** commands.

smtpd_recipient_restrictions
Restrict what recipient addresses are allowed in **RCPT TO** commands.

smtpd_etrn_restrictions
Restrict what domain names can be used in **ETRN** commands, and what clients may issue **ETRN** commands.

smtpd_data_restrictions
Restrictions on the **DATA** command. Currently, the only restriction that makes sense here is **reject_unauth_pipelining**.

allow_untrusted_routing
Allow untrusted clients to specify addresses with sender-specified routing. Enabling this opens up nasty relay loopholes involving trusted backup MX hosts.

smtpd_restriction_classes
Declares the name of zero or more parameters that contain a list of UCE restrictions. The names of these parameters can then be used instead of the restriction lists that they represent.

smtpd_null_access_lookup_key
The lookup key to be used in SMTPD access tables instead of the null sender address. A null sender address cannot be looked up.

maps_rbl_domains (deprecated)
List of DNS domains that publish the addresses of blacklisted hosts. This is used with the deprecated **reject_maps_rbl** restriction.

permit_mx_backup_networks
Only domains whose primary MX hosts match the listed networks are eligible for the **permit_mx_backup** feature.

relay_domains
Restrict what domains this mail system will relay mail to. The domains are routed to the delivery agent specified with the **relay_transport** setting.

UCE control responses

access_map_reject_code
Response code when a client violates an access database restriction.

default_rbl_reply

Default template reply when a request is RBL blacklisted. This template is used by the **reject_rbl_*** and **reject_rhsbl_*** restrictions. See also: **rbl_reply_maps** and **smtpd_expansion_filter**.

defer_code
Response code when a client request is rejected by the **defer** restriction.

invalid_hostname_reject_code
Response code when a client violates the **reject_invalid_hostname** restriction.

maps_rbl_reject_code
Response code when a request is RBL blacklisted.

rbl_reply_maps
Table with template responses for RBL blacklisted requests, indexed by RBL domain name. These templates are used by the **reject_rbl_*** and **reject_rhsbl_*** restrictions. See also: **default_rbl_reply** and **smtpd_expansion_filter**.

reject_code
Response code when the client matches a **reject** restriction.

relay_domains_reject_code
Response code when a client attempts to violate the mail relay policy.

unknown_address_reject_code
Response code when a client violates the **reject_unknown_address** restriction.

unknown_client_reject_code
Response code when a client without address to name mapping violates the **reject_unknown_client** restriction.

unknown_hostname_reject_code
Response code when a client violates the **reject_unknown_hostname** restriction.

SEE ALSO

trivial-rewrite(8) address resolver
cleanup(8) message canonicalization
master(8) process manager
syslogd(8) system logging

LICENSE

The Secure Mailer license must be distributed with this software.

PICKUP(8) PICKUP(8)

NAME

pickup - Postfix local mail pickup

SYNOPSIS

pickup [generic Postfix daemon options]

DESCRIPTION

The **pickup** daemon waits for hints that new mail has been dropped into the **maildrop** directory, and feeds it into the **cleanup**(8) daemon. Ill-formatted files are deleted without notifying the originator. This program expects to be run from the **master**(8) process manager.

STANDARDS

None. The **pickup** daemon does not interact with the outside world.

SECURITY

The **pickup** daemon is moderately security sensitive. It runs with fixed low privilege and can run in a chrooted environment. However, the program reads files from potentially hostile users. The **pickup** daemon opens no files for writing, is careful about what files it opens for reading, and does not actually touch any data that is sent to its public service endpoint.

DIAGNOSTICS

Problems and transactions are logged to **syslogd**(8).

BUGS

The **pickup** daemon copies mail from file to the **cleanup**(8) daemon. It could avoid message copying overhead by sending a file descriptor instead of file data, but then the already complex **cleanup**(8) daemon would have to deal with unfiltered user data.

CONFIGURATION PARAMETERS

The following **main.cf** parameters are especially relevant to this program. See the Postfix **main.cf** file for syntax details and for default values. Use the **postfix reload** command after a configuration change.

Content inspection controls

content_filter

The name of a mail delivery transport that filters mail and that either bounces mail or re-injects the result back into Postfix. This parameter uses the same syntax as the right-hand side of a Postfix transport table.

Miscellaneous

always_bcc

Address to send a copy of each message that enters the system.

queue_directory

Top-level directory of the Postfix queue.

SEE ALSO

cleanup(8) message canonicalization
master(8) process manager
sendmail(1), postdrop(8) mail posting agent
syslogd(8) system logging

TRIVIAL-REWRITE(8) TRIVIAL-REWRITE(8)

NAME

trivial-rewrite - Postfix address rewriting and resolving daemon

SYNOPSIS

trivial-rewrite [generic Postfix daemon options]

DESCRIPTION

The **trivial-rewrite** daemon processes two types of client service requests:

rewrite

Rewrite an address to standard form. The **trivial-rewrite** daemon by default appends local domain information to unqualified addresses, swaps bang paths to domain form, and strips source routing information. This process is under control of several configuration parameters (see below).

resolve

Resolve an address to a (*transport*, *nexthop*, *recipient*) triple. The meaning of the results is as follows:

transport

The delivery agent to use. This is the first field of an entry in the **master.cf** file.

nexthop

The host to send to and optional delivery method information.

recipient

The envelope recipient address that is passed on to *nexthop*.

DEFAULT DELIVERY METHODS

By default, Postfix uses one of the following delivery methods. This may be overruled with the optional transport(5) table. The default delivery method is selected by matching the recipient address domain against one of the following:

$mydestination

$inet_interfaces

The transport and optional nexthop are specified with **$local_transport**. The default nexthop is the recipient domain.

$virtual_alias_domains

The recipient address is undeliverable (user unknown). By definition, all known addresses in a virtual alias domain are aliased to other addresses.

$virtual_mailbox_domains

The transport and optional nexthop are specified with **$virtual_transport**. The default nexthop is

the recipient domain.

$relay_domains
The transport and optional nexthop are specified with **$relay_transport**. This overrides the optional nexthop information that is specified with **$relayhost**. The default nexthop is the recipient domain.

none of the above
The transport and optional nexthop are specified with **$default_transport**. This overrides the optional nexthop information that is specified with **$relayhost**. The default nexthop is the recipient domain.

SERVER PROCESS MANAGEMENT
The trivial-rewrite servers run under control by the Postfix master server. Each server can handle multiple simultaneous connections. When all servers are busy while a client connects, the master creates a new server process, provided that the trivial-rewrite server process limit is not exceeded. Each trivial-rewrite server terminates after serving at least **$max_use** clients of after **$max_idle** seconds of idle time.

STANDARDS
None. The command does not interact with the outside world.

SECURITY
The **trivial-rewrite** daemon is not security sensitive. By default, this daemon does not talk to remote or local users. It can run at a fixed low privilege in a chrooted environment.

DIAGNOSTICS
Problems and transactions are logged to **syslogd**(8).

BUGS
CONFIGURATION PARAMETERS
The following **main.cf** parameters are especially relevant to this program. See the Postfix **main.cf** file for syntax details and for default values. Use the **postfix reload** command after a configuration change.

Miscellaneous
empty_address_recipient
The recipient that is substituted for the null address.

inet_interfaces
The network interfaces that this mail system receives mail on. This information is used to determine if *user*@[*net.work.addr.ess*] is local or remote. Mail for local users is given to the **$local_transport**.

mydestination
List of domains that are given to the **$local_transport**.

virtual_alias_domains

List of simulated virtual domains (domains with all recipients aliased to some other local or remote domain).

virtual_mailbox_domains
List of domains that are given to the **$virtual_transport**.

relay_domains
List of domains that are given to the **$relay_transport**.

resolve_unquoted_address
When resolving an address, do not quote the address localpart as per RFC 822, so that additional **@**, **%** or **!** characters remain visible. This is technically incorrect, but allows us to stop relay attacks when forwarding mail to a Sendmail primary MX host.

relocated_maps
Tables with contact information for users, hosts or domains that no longer exist. See **relocated**(5).

Rewriting

myorigin
The domain that locally-posted mail appears to come from.

allow_percent_hack
Rewrite *user%domain* to *user@domain*.

append_at_myorigin
Rewrite *user* to *user*@**$myorigin**.

append_dot_mydomain
Rewrite *user@host* to *user@host*.**$mydomain**.

swap_bangpath
Rewrite *site*!*user* to *user@site*.

Routing

local_transport
Where to deliver mail for destinations that match **$mydestination** or **$inet_interfaces**. The default transport is **local:$myhostname**.

Syntax is *transport*:*nexthop*; see **transport**(5) for details. The :*nexthop* part is optional.

virtual_transport
Where to deliver mail for non-local domains that match **$virtual_mailbox_domains**. The default transport is **virtual**.

Syntax is *transport*:*nexthop*; see **transport**(5) for details. The :*nexthop* part is optional.

relay_transport
Where to deliver mail for non-local domains that match **$relay_domains**. The default transport is **relay** (which normally is a clone of the **smtp** transport).

Syntax is *transport*:*nexthop*; see **transport**(5) for details. The :*nexthop* part is optional.

default_transport
Where to deliver all other non-local mail. The default transport is **smtp**.

Syntax is *transport*:*nexthop*; see **transport**(5) for details. The :*nexthop* part is optional.

parent_domain_matches_subdomains
List of Postfix features that use *domain.tld* patterns to match *sub.domain.tld* (as opposed to requiring *.domain.tld* patterns).

relayhost
The default host to send non-local mail to when no host is specified with **$relay_transport** or **$default_transport**, and when the recipient address does not match the optional the **transport**(5) table.

transport_maps
List of tables with *recipient* or *domain* to (*transport*, *nexthop*) mappings.

SEE ALSO
master(8) process manager
syslogd(8) system logging
transport(5) transport table format
relocated(5) format of the "user has moved" table

LICENSE
The Secure Mailer license must be distributed with this software.

AUTHOR(S)
Wietse Venema
IBM T.J. Watson Research
P.O. Box 704
Yorktown Heights, NY 10598, USA

TRIVIAL-REWRITE(8)

CLEANUP(8) CLEANUP(8)

NAME

cleanup - canonicalize and enqueue Postfix message

SYNOPSIS

cleanup [generic Postfix daemon options]

DESCRIPTION

The **cleanup** daemon processes inbound mail, inserts it into the **incoming** mail queue, and informs the queue manager of its arrival.

The **cleanup** daemon always performs the following transformations:

- Insert missing message headers: (**Resent-**) **From:**, **To:**, **Message-Id:**, and **Date:**.
- Extract envelope recipient addresses from (**Resent-**) **To:**, **Cc:** and **Bcc:** message headers when no recipients are specified in the message envelope.
- Transform envelope and header addresses to the standard *user@fully-qualified-domain* form that is expected by other Postfix programs. This task is delegated to the **trivial-rewrite**(8) daemon.
- Eliminate duplicate envelope recipient addresses.

The following address transformations are optional:

- Optionally, rewrite all envelope and header addresses according to the mappings specified in the **canonical**(5) lookup tables.
- Optionally, masquerade envelope sender addresses and message header addresses (i.e. strip host or domain information below all domains listed in the **masquerade_domains** parameter, except for user names listed in **masquerade_exceptions**). By default, address masquerading does not affect envelope recipients.
- Optionally, expand envelope recipients according to information found in the **virtual**(5) lookup tables.

The **cleanup** daemon performs sanity checks on the content of each message. When it finds a problem, by default it returns a diagnostic status to the client, and leaves it up to the client to deal with the problem. Alternatively, the client can request the **cleanup** daemon to bounce the message back to the sender in case of trouble.

DIAGNOSTICS

Problems and transactions are logged to **syslogd**(8).

BUGS

Table-driven rewriting rules make it hard to express **if then else** and other logical relationships.

CONFIGURATION PARAMETERS

The following **main.cf** parameters are especially relevant to this program. See the Postfix **main.cf** file for syntax details and for default values. Use the **postfix reload** command after a configuration change.

Content filtering

body_checks

Lookup tables with content filters for message body lines. These filters see physical lines one at a time, in chunks of at most line_length_limit bytes.

body_checks_size_limit

The amount of content per message body segment that is subjected to **$body_checks** filtering.

header_checks

mime_header_checks (default: **$header_checks**)

nested_header_checks (default: **$header_checks**)

Lookup tables with content filters for message header lines: respectively, these are applied to the primary message headers (not including MIME headers), to the MIME headers anywhere in the message, and to the initial headers of attached messages. These filters see logical headers one at a time, including headers that span multiple lines.

MIME Processing

disable_mime_input_processing

While receiving, give no special treatment to **Content-Type:** message headers; all text after the initial message headers is considered to be part of the message body.

mime_boundary_length_limit

The amount of space that will be allocated for MIME multipart boundary strings. The MIME processor is unable to distinguish between boundary strings that do not differ in the first **$mime_boundary_length_limit** characters.

mime_nesting_limit

The maximal nesting level of multipart mail that the MIME processor can handle. Refuse mail that is nested deeper.

strict_8bitmime

Turn on both **strict_7bit_headers** and **strict_8bitmime_body**.

strict_7bit_headers

Reject mail with 8-bit text in message headers. This blocks mail from poorly written applications.

strict_8bitmime_body

Reject mail with 8-bit text in content that claims

to be 7-bit, or in content that has no explicit content encoding information. This blocks mail from poorly written mail software. Unfortunately, this also breaks majordomo approval requests when the included request contains valid 8-bit MIME mail, and it breaks bounces from mailers that do not properly encapsulate 8-bit content (for example, bounces from qmail or from old versions of Postfix).

strict_mime_domain_encoding
Reject mail with invalid **Content-Transfer-Encoding:** information for message/* or multipart/*. This blocks mail from poorly written software.

Miscellaneous

always_bcc
Address to send a copy of each message that enters the system.

hopcount_limit
Limit the number of **Received:** message headers.

undisclosed_recipients_header
The header line that is inserted when no recipients were specified in (Resent-)To: or (Resent-)Cc: message headers.

Address transformations

empty_address_recipient
The destination for undeliverable mail from <>. This substitution is done before all other address rewriting.

canonical_maps
Address mapping lookup table for sender and recipient addresses in envelopes and headers.

recipient_canonical_maps
Address mapping lookup table for envelope and header recipient addresses.

sender_canonical_maps
Address mapping lookup table for envelope and header sender addresses.

masquerade_classes
List of address classes subject to masquerading: zero or more of **envelope_sender**, **envelope_recipient**, **header_sender**, **header_recipient**.

masquerade_domains
List of domains that hide their subdomain structure.

masquerade_exceptions
List of user names that are not subject to address masquerading.

virtual_alias_maps
Address mapping lookup table for envelope recipient addresses.

Resource controls

duplicate_filter_limit
Limits the number of envelope recipients that are remembered.

header_address_token_limit
Limits the number of address tokens used to process a message header.

header_size_limit
Limits the amount of memory in bytes used to process a message header.

in_flow_delay
Amount of time to pause before accepting a message, when the message arrival rate exceeds the message delivery rate.

extract_recipient_limit
Limit the amount of recipients extracted from message headers.

SEE ALSO

canonical(5) canonical address lookup table format
qmgr(8) queue manager daemon
syslogd(8) system logging
trivial-rewrite(8) address rewriting
virtual(5) virtual alias lookup table format

FILES

/etc/postfix/canonical*, canonical mapping table
/etc/postfix/virtual*, virtual mapping table

LICENSE

The Secure Mailer license must be distributed with this software.

AUTHOR(S)

Wietse Venema
IBM T.J. Watson Research
P.O. Box 704
Yorktown Heights, NY 10598, USA

CLEANUP(8)

Look-up tables under Postfix

```
# ACCESS(5)                                                ACCESS(5)
#
# NAME
#       access - format of Postfix access table
#
# SYNOPSIS
#       postmap /etc/postfix/access
#
# DESCRIPTION
#       The  optional access table directs the Postfix SMTP server
#       to selectively  reject  or  accept  mail.  Access  can  be
#       allowed  or  denied for specific host names, domain names,
#       networks, host network addresses or mail addresses.
#
#       Normally, the access table is specified  as  a  text  file
#       that  serves  as  input  to  the  postmap(1) command.  The
#       result, an indexed file in dbm or db format, is  used  for
#       fast  searching  by  the  mail system. Execute the command
#       postmap  /etc/postfix/access  in  order  to  rebuild   the
#       indexed file after changing the access table.
#
#       When  the  table  is provided via other means such as NIS,
#       LDAP or SQL, the same lookups are  done  as  for  ordinary
#       indexed files.
#
#       Alternatively,  the  table  can  be provided as a regular-
#       expression map where patterns are given as regular expres-
#       sions.  In  that  case, the lookups are done in a slightly
#       different way as described below.
#
# TABLE FORMAT
#       The format of the access table is as follows:
#
#       pattern action
#              When pattern matches a mail address, domain or host
#              address, perform the corresponding action.
#
#       blank lines and comments
#              Empty  lines and whitespace-only lines are ignored,
#              as are lines whose first  non-whitespace  character
#              is a `#'.
#
#       multi-line text
#              A  logical  line starts with non-whitespace text. A
#              line that starts with whitespace continues a  logi-
#              cal line.
#
# EMAIL ADDRESS PATTERNS
#       With lookups from indexed files such as DB or DBM, or from
#       networked tables such as NIS, LDAP or SQL,  the  following
#       lookup patterns are examined in the order as listed:
#
#       user@domain
#              Matches the specified mail address.
#
#       domain.tld
#              Matches  domain.tld  as the domain part of an email
#              address.
#
#              The pattern domain.tld also matches subdomains, but
#              only when the string smtpd_access_maps is listed in
#              the Postfix  parent_domain_matches_subdomains  con-
#              figuration setting.  Otherwise, specify .domain.tld
#              (note the initial dot) in  order  to  match  subdo-
#              mains.
#
#       user@  Matches  all mail addresses with the specified user
```

```
#               part.
#
#       Note: lookup of the null sender address  is  not  possible
#       with  some types of lookup table. By default, Postfix uses
#       <> as the lookup key for  such  addresses.  The  value  is
#       specified    with    the    workaround    is    to    specify
#       smtpd_null_access_lookup_key  parameter  in  the   Postfix
#       main.cf file.
#
# ADDRESS EXTENSION
#       When a mail address localpart contains the optional recip-
#       ient delimiter (e.g., user+foo@domain), the  lookup  order
#       becomes:  user+foo@domain, user@domain, domain, user+foo@,
#       and user@.
#
# HOST NAME/ADDRESS PATTERNS
#       With lookups from indexed files such as DB or DBM, or from
#       networked  tables  such as NIS, LDAP or SQL, the following
#       lookup patterns are examined in the order as listed:
#
#       domain.tld
#               Matches domain.tld.
#
#               The pattern domain.tld also matches subdomains, but
#               only when the string smtpd_access_maps is listed in
#               the Postfix  parent_domain_matches_subdomains  con-
#               figuration setting.  Otherwise, specify .domain.tld
#               (note the initial dot) in  order  to  match  subdo-
#               mains.
#
#       net.work.addr.ess
#
#       net.work.addr
#
#       net.work
#
#       net     Matches  any host address in the specified network.
#               A network address is a  sequence  of  one  or  more
#               octets separated by ".".
#
# ACTIONS
#       [45]NN text
#               Reject  the  address etc. that matches the pattern,
#               and respond with the numerical code and text.
#
#       REJECT Reject the address etc. that matches the pattern. A
#               generic error response message is generated.
#
#       OK      Accept the address etc. that matches the pattern.
#
#       all-numerical
#               An all-numerical result is treated as OK. This for-
#               mat is generated by address-based relay  authoriza-
#               tion schemes.
#

#       restriction...
#               Apply the named UCE restriction(s) (permit, reject,
#               reject_unauth_destination, and so on).
#
# REGULAR EXPRESSION TABLES
#       This section describes how the table lookups  change  when
#       the table is given in the form of regular expressions. For
#       a description of regular expression lookup  table  syntax,
#       see regexp_table(5) or pcre_table(5).
#
#       Each  pattern  is  a regular expression that is applied to
#       the entire string being looked up. Depending on the appli-
#       cation,  that  string  is  an  entire  client hostname, an
#       entire client IP address, or an entire mail address. Thus,
```

```
#       no  parent  domain  or  parent  network  search  is  done,
#       user@domain mail addresses are not broken  up  into  their
#       user@ and domain constituent parts, nor is user+foo broken
#       up into user and foo.
#
#       Patterns are applied in the  order  as  specified  in  the
#       table,  until  a  pattern is found that matches the search
#       string.
#
#       Actions are the same as with indexed  file  lookups,  with
#       the  additional feature that parenthesized substrings from
#       the pattern can be interpolated as $1, $2 and so on.
#
# BUGS
#       The table format does not understand quoting  conventions.
#
# SEE ALSO
#       postmap(1) create mapping table
#       smtpd(8) smtp server
#       pcre_table(5) format of PCRE tables
#       regexp_table(5) format of POSIX regular expression tables
#
# LICENSE
#       The  Secure  Mailer  license must be distributed with this
#       software.
#
# AUTHOR(S)
#       Wietse Venema
#       IBM T.J. Watson Research
#       P.O. Box 704
#       Yorktown Heights, NY 10598, USA
#
#
```

ALIASES(5) ALIASES(5)

NAME

aliases - format of the Postfix alias database

SYNOPSIS

newaliases

DESCRIPTION

The **aliases** table provides a system-wide mechanism to redirect mail for local recipients. The redirections are processed by the Postfix **local**(8) delivery agent.

Normally, the **aliases** table is specified as a text file that serves as input to the **postalias**(1) command. The result, an indexed file in **dbm** or **db** format, is used for fast lookup by the mail system. Execute the command **newaliases** in order to rebuild the indexed file after changing the Postfix alias database.

The input and output file formats are expected to be compatible with Sendmail version 8, and are expected to be suitable for the use as NIS maps.

Users can control delivery of their own mail by setting up **.forward** files in their home directory. Lines in per-user **.forward** files have the same syntax as the right-hand side of **aliases** entries.

The format of the alias database input file is as follows:

- An alias definition has the form

  name: *value1*, *value2*, ...

- Empty lines and whitespace-only lines are ignored, as are lines whose first non-whitespace character is a `#'.

- A logical line starts with non-whitespace text. A line that starts with whitespace continues a logical line.

The *name* is a local address (no domain part). Use double quotes when the name contains any special characters such as whitespace, `#', `:', or `@'. The *name* is folded to lowercase, in order to make database lookups case insensitive.

In addition, when an alias exists for **owner-***name*, delivery diagnostics are directed to that address, instead of to the originator. This is typically used to direct delivery errors to the owner of a mailing list, who is in a better position to deal with mailing list delivery problems than the originator of the undelivered mail.

The *value* contains one or more of the following:

address

Mail is forwarded to *address*, which is compatible with the RFC 822 standard.

/file/name
Mail is appended to */file/name*. See **local**(8) for details of delivery to file. Delivery is not limited to regular files. For example, to dispose of unwanted mail, deflect it to **/dev/null**.

|*command*
Mail is piped into *command*. Commands that contain special characters, such as whitespace, should be enclosed between double quotes. See **local**(8) for details of delivery to command.

When the command fails, a limited amount of command output is mailed back to the sender. The file **/usr/include/sysexits.h** defines the expected exit status codes. For example, use **|"exit 67"** to simulate a "user unknown" error, and **|"exit 0"** to implement an expensive black hole.

:include:*/file/name*
Mail is sent to the destinations listed in the named file. Lines in **:include:** files have the same syntax as the right-hand side of alias entries.

A destination can be any destination that is described in this manual page. However, delivery to "|*command*" and */file/name* is disallowed by default. To enable, edit the **allow_mail_to_commands** and **allow_mail_to_files** configuration parameters.

ADDRESS EXTENSION
When alias database search fails, and the recipient local-part contains the optional recipient delimiter (e.g., *user+foo*), the search is repeated for the unextended address (e.g., *user*).

CONFIGURATION PARAMETERS
The following **main.cf** parameters are especially relevant to this topic. See the Postfix **main.cf** file for syntax details and for default values. Use the **postfix reload** command after a configuration change.

alias_maps
List of alias databases.

allow_mail_to_commands
Restrict the usage of mail delivery to external command.

allow_mail_to_files
Restrict the usage of mail delivery to external file.

expand_owner_alias
When delivering to an alias that has an **owner-** companion alias, set the envelope sender address to the right-hand side of the owner alias, instead using of the left-hand side address.

owner_request_special
Give special treatment to **owner-***xxx* and *xxx***-request** addresses.

recipient_delimiter
Delimiter that separates recipients from address extensions.

BUGS

Regular expression alias lookup tables are allowed, but substitution of $1 etc. is forbidden because that would open a security loophole.

STANDARDS

RFC 822 (ARPA Internet Text Messages)

SEE ALSO

local(8) local delivery agent
newaliases(1) alias database management
regexp_table(5) POSIX regular expression table format
pcre_table(5) Perl Compatible Regular Expression table format

LICENSE

The Secure Mailer license must be distributed with this software.

AUTHOR(S)

Wietse Venema
IBM T.J. Watson Research
P.O. Box 704
Yorktown Heights, NY 10598, USA

ALIASES(5)

```
# CANONICAL(5)                                          CANONICAL(5)
#
# NAME
#       canonical - format of Postfix canonical table
#
# SYNOPSIS        postmap /etc/postfix/canonical
#
# DESCRIPTION
#       The  optional canonical table specifies an address mapping
#       for local and non-local addresses. The mapping is used  by
#       the  cleanup(8) daemon.  The address mapping is recursive.
#
#       Normally, the canonical table is specified as a text  file
#       that  serves  as  input  to  the  postmap(1) command.  The
#       result, an indexed file in dbm or db format, is  used  for
#       fast  searching  by  the  mail system. Execute the command
#       postmap /etc/postfix/canonical in  order  to  rebuild  the
#       indexed file after changing the text file.
#
#       When  the  table  is provided via other means such as NIS,
#       LDAP or SQL, the same lookups are  done  as  for  ordinary
#       indexed files.
#
#       Alternatively,  the  table  can  be provided as a regular-
#       expression map where patterns are given as regular expres-
#       sions.  In  that  case, the lookups are done in a slightly
#       different way as described below.
#
#       The  canonical  mapping  affects  both  message  header
#       addresses (i.e. addresses that appear inside messages) and
#       message envelope addresses  (for  example,  the  addresses
#       that  are  used in SMTP protocol commands). Think Sendmail
#       rule set S3, if you like.
#
#       Typically, one would use the canonical  table  to  replace
#       login   names   by  Firstname.Lastname,  or  to  clean  up
#       addresses produced by legacy mail systems.
#
#       The canonical mapping is not to be confused  with  virtual
#       domain support. Use the virtual(5) map for that purpose.
#
#       The  canonical  mapping  is  not to be confused with local
#       aliasing.  Use the aliases(5) map for that purpose.
#
# TABLE FORMAT
#       The format of the canonical table is as follows:
#
#       pattern result
#              When pattern matches a mail address, replace it  by
#              the corresponding result.
#
#       blank lines and comments
#              Empty  lines and whitespace-only lines are ignored,
#              as are lines whose first  non-whitespace  character
#              is a `#'.
#
#       multi-line text
#              A  logical  line starts with non-whitespace text. A
#              line that starts with whitespace continues a  logi-
#              cal line.
#
```

```
#       With lookups from indexed files such as DB or DBM, or from
#       networked tables such as NIS, LDAP or  SQL,  patterns  are
#       tried in the order as listed below:
#       user@domain address
#              user@domain  is  replaced by address. This form has
#              the highest precedence.
#
#              This form useful to clean up addresses produced  by
#              legacy  mail  systems.  It can also be used to pro-
#              duce Firstname.Lastname style  addresses,  but  see
#              below for a simpler solution.
#
#       user address
#              user@site is replaced by address when site is equal
#              to $myorigin, when site is  listed  in  $mydestina-
#              tion, or when it is listed in $inet_interfaces.
#
#              This  form  is  useful for replacing login names by
#              Firstname.Lastname.
#
#       @domain address
#              Every address in domain  is  replaced  by  address.
#              This form has the lowest precedence.
#
#       In  all the above forms, when address has the form @other-
#       domain, the result is the same user in otherdomain.
#
# ADDRESS EXTENSION
#       When a mail address localpart contains the optional recip-
#       ient  delimiter  (e.g., user+foo@domain), the lookup order
#       becomes: user+foo@domain, user@domain, user+foo, user, and
#       @domain.   An unmatched address extension (+foo) is propa-
#       gated to the result of table lookup.
#
# REGULAR EXPRESSION TABLES
#       This section describes how the table lookups  change  when
#       the table is given in the form of regular expressions. For
#       a description of regular expression lookup  table  syntax,
#       see regexp_table(5) or pcre_table(5).
#
#       Each  pattern  is  a regular expression that is applied to
#       the entire address being looked up. Thus, user@domain mail
#       addresses  are  not  broken up into their user and @domain
#       constituent parts, nor is user+foo broken up into user and
#       foo.
#
#       Patterns  are  applied  in  the  order as specified in the
#       table, until a pattern is found that  matches  the  search
#       string.
#
#       Results  are  the  same as with indexed file lookups, with
#       the additional feature that parenthesized substrings  from
#       the pattern can be interpolated as $1, $2 and so on.
#
# BUGS
#       The  table format does not understand quoting conventions.
#
# CONFIGURATION PARAMETERS
#       The following main.cf parameters are  especially  relevant
#       to  this  topic.  See  the Postfix main.cf file for syntax
#       details and for default values.  Use  the  postfix  reload
#       command after a configuration change.
#       canonical_maps
#              List of canonical mapping tables.
#

#       recipient_canonical_maps
#              Address  mapping  lookup  table  for  envelope  and
#              header recipient addresses.
#
```

```
#       sender_canonical_maps
#              Address  mapping  lookup  table  for  envelope  and
#              header sender addresses.
#
#       Other parameters of interest:
#
#       inet_interfaces
#              The  network  interface  addresses that this system
#              receives mail on.
#
#       masquerade_classes
#              List of address classes  subject  to  masquerading:
#              zero  or  more of envelope_sender, envelope_recipi-
#              ent, header_sender, header_recipient.
#
#       masquerade_domains
#              List of domains that hide  their  subdomain  struc-
#              ture.
#
#       masquerade_exceptions
#              List  of user names that are not subject to address
#              masquerading.
#
#       mydestination
#              List of domains that  this  mail  system  considers
#              local.
#
#       myorigin
#              The domain that is appended to locally-posted mail.
#
#       owner_request_special
#              Give special treatment to owner-xxx and xxx-request
#              addresses.
#
# SEE ALSO
#       cleanup(8) canonicalize and enqueue mail
#       postmap(1) create mapping table
#       virtual(5) virtual domain mapping
#       pcre_table(5) format of PCRE tables
#       regexp_table(5) format of POSIX regular expression tables
#
# LICENSE
#       The  Secure  Mailer  license must be distributed with this
#       software.
#
# AUTHOR(S)
#       Wietse Venema
#       IBM T.J. Watson Research
#       P.O. Box 704
#       Yorktown Heights, NY 10598, USA
#
#                                                                1
```

NAME

canonical - format of Postfix canonical table

SYNOPSIS

postmap /etc/postfix/canonical

postmap -q "*string*" **/etc/postfix/canonical**

postmap -q - /etc/postfix/canonical <*inputfile*

DESCRIPTION

The optional **canonical** table specifies an address mapping for local and non-local addresses. The mapping is used by the **cleanup**(8) daemon. The address mapping is recursive.

Normally, the **canonical** table is specified as a text file that serves as input to the **postmap**(1) command. The result, an indexed file in **dbm** or **db** format, is used for fast searching by the mail system. Execute the command **postmap /etc/postfix/canonical** in order to rebuild the indexed file after changing the text file.

When the table is provided via other means such as NIS, LDAP or SQL, the same lookups are done as for ordinary indexed files.

Alternatively, the table can be provided as a regular-expression map where patterns are given as regular expressions. In that case, the lookups are done in a slightly different way as described below.

The **canonical** mapping affects both message header addresses (i.e. addresses that appear inside messages) and message envelope addresses (for example, the addresses that are used in SMTP protocol commands). Think Sendmail rule set **S3**, if you like.

Typically, one would use the **canonical** table to replace login names by *Firstname.Lastname*, or to clean up addresses produced by legacy mail systems.

The **canonical** mapping is not to be confused with *virtual domain* support. Use the **virtual**(5) map for that purpose.

The **canonical** mapping is not to be confused with local aliasing. Use the **aliases**(5) map for that purpose.

TABLE FORMAT

The format of the **canonical** table is as follows:

pattern result

When *pattern* matches a mail address, replace it by the corresponding *result*.

blank lines and comments

Empty lines and whitespace-only lines are ignored, as are lines whose first non-whitespace character is a `#'.

multi-line text
A logical line starts with non-whitespace text. A line that starts with whitespace continues a logical line.

With lookups from indexed files such as DB or DBM, or from networked tables such as NIS, LDAP or SQL, patterns are tried in the order as listed below:

user@domain address
user@domain is replaced by *address*. This form has the highest precedence.

This is useful to clean up addresses produced by legacy mail systems. It can also be used to produce *Firstname.Lastname* style addresses, but see below for a simpler solution.

user address
user@site is replaced by *address* when *site* is equal to $**myorigin**, when *site* is listed in $**mydestination**, or when it is listed in $**inet_interfaces**.

This form is useful for replacing login names by *Firstname.Lastname*.

@domain address
Every address in *domain* is replaced by *address*. This form has the lowest precedence.

In all the above forms, when *address* has the form *@otherdomain*, the result is the same user in *otherdomain*.

ADDRESS EXTENSION

When a mail address localpart contains the optional recipient delimiter (e.g., *user+foo@domain*), the lookup order becomes: *user+foo@domain*, *user@domain*, *user+foo*, *user*, and *@domain*. An unmatched address extension (*+foo*) is propagated to the result of table lookup.

REGULAR EXPRESSION TABLES

This section describes how the table lookups change when the table is given in the form of regular expressions. For a description of regular expression lookup table syntax, see **regexp_table**(5) or **pcre_table**(5).

Each pattern is a regular expression that is applied to the entire address being looked up. Thus, *user@domain* mail addresses are not broken up into their *user* and *@domain* constituent parts, nor is *user+foo* broken up into *user* and *foo*.

Patterns are applied in the order as specified in the table, until a pattern is found that matches the search string.

Results are the same as with indexed file lookups, with the additional feature that parenthesized substrings from the pattern can be interpolated as **$1**, **$2** and so on.

BUGS

The table format does not understand quoting conventions.

CONFIGURATION PARAMETERS

The following **main.cf** parameters are especially relevant to this topic. See the Postfix **main.cf** file for syntax details and for default values. Use the **postfix reload** command after a configuration change.

canonical_maps
List of canonical mapping tables.

recipient_canonical_maps
Address mapping lookup table for envelope and header recipient addresses.

sender_canonical_maps
Address mapping lookup table for envelope and header sender addresses.

Other parameters of interest:

inet_interfaces
The network interface addresses that this system receives mail on. You need to stop and start Postfix when this parameter changes.

masquerade_classes
List of address classes subject to masquerading: zero or more of **envelope_sender**, **envelope_recipient**, **header_sender**, **header_recipient**.

masquerade_domains
List of domains that hide their subdomain structure.

masquerade_exceptions
List of user names that are not subject to address masquerading.

mydestination
List of domains that this mail system considers local.

myorigin
The domain that is appended to locally-posted mail.

owner_request_special
Give special treatment to **owner-***xxx* and *xxx*-**request** addresses.

SEE ALSO

cleanup(8) canonicalize and enqueue mail
postmap(1) create mapping table
virtual(5) virtual domain mapping
pcre_table(5) format of PCRE tables
regexp_table(5) format of POSIX regular expression tables

LICENSE

The Secure Mailer license must be distributed with this software.

AUTHOR(S)

Wietse Venema

```
# RELOCATED(5)                                          RELOCATED(5)
#
# NAME
#       relocated - format of Postfix relocated table
#
# SYNOPSIS
#       postmap /etc/postfix/relocated
#
# DESCRIPTION
#       The optional relocated table provides the information that
#       is used in "user has moved to  new_location"  bounce  mes-
#       sages.
#
#       Normally,  the relocated table is specified as a text file
#       that serves as  input  to  the  postmap(1)  command.   The
#       result,  an  indexed file in dbm or db format, is used for
#       fast searching by the mail  system.  Execute  the  command
#       postmap  /etc/postfix/relocated  in  order  to rebuild the
#       indexed file after changing the relocated table.
#
#       When the table is provided via other means  such  as  NIS,
#       LDAP  or  SQL,  the  same lookups are done as for ordinary
#       indexed files.
#
#       Alternatively, the table can be  provided  as  a  regular-
#       expression map where patterns are given as regular expres-
#       sions. In that case, the lookups are done  in  a  slightly
#       different way as described below.
#
#       Table lookups are case insensitive.
#
# TABLE FORMAT
#       The format of the table is as follows:
#
#       o      An entry has one of the following form:
#                   key  new_location
#              Where  new_location  specifies  contact information
#              such as an  email  address,  or  perhaps  a  street
#              address or telephone number.
#
#       o      Empty  lines and whitespace-only lines are ignored,
#              as are lines whose first  non-whitespace  character
#              is a `#'.
#
#       o      A  logical  line starts with non-whitespace text. A
#              line that starts with whitespace continues a  logi-
#              cal line.
#
#       With lookups from indexed files such as DB or DBM, or from
#       networked tables such as NIS, LDAP or SQL, the  key  field
#       is one of the following:
#
#       user@domain
#              Matches  user@domain. This form has precedence over
#              all other forms.
#
#       user   Matches user@site when site is $myorigin, when site
#              is listed in $mydestination, or when site is listed
#              in $inet_interfaces.
#
```

```
#        @domain
#               Matches every address in domain. This form has  the
#               lowest precedence.
#
# ADDRESS EXTENSION
#        When a mail address localpart contains the optional recip-
#        ient delimiter (e.g., user+foo@domain), the  lookup  order
#        becomes: user+foo@domain, user@domain, user+foo, user, and
#        @domain.
#
```

```
# TRANSPORT(5)                                          TRANSPORT(5)
#
# NAME
#        transport - format of Postfix transport table
#
# SYNOPSIS
#        postmap /etc/postfix/transport
#
# DESCRIPTION
#        The  optional  transport  table  specifies  a mapping from
#        domain hierarchies to message delivery  transports  and/or
#        relay hosts. The mapping is used by the trivial-rewrite(8)
#        daemon.
#
#        Normally, the transport table is specified as a text  file
#        that  serves  as  input  to  the  postmap(1) command.  The
#        result, an indexed file in dbm or db format, is  used  for
#        fast  searching  by  the  mail system. Execute the command
#        postmap /etc/postfix/transport in  order  to  rebuild  the
#        indexed file after changing the transport table.
#        When  the  table  is provided via other means such as NIS,
#        LDAP or SQL, the same lookups are  done  as  for  ordinary
#        indexed files.
#
#        Alternatively,  the  table  can  be provided as a regular-
#        expression map where patterns are given as regular expres-
#        sions.  In  that  case, the lookups are done in a slightly
#        different way as described below.
#
# TABLE FORMAT
#        The format of the transport table is as follows:
#
#        pattern result
#               When pattern matches the  domain,  use  the  corre-
#               sponding result.  A pattern of `*' matches all
#               entries.
#
#        blank lines and comments
#               Empty  lines and whitespace-only lines are ignored,
#               as are lines whose first  non-whitespace  character
#               is a `#'.
#
#        multi-line text
#               A  logical  line starts with non-whitespace text. A
#               line that starts with whitespace continues a  logi-
#               cal line.
#
#        With lookups from indexed files such as DB or DBM, or from
#        networked tables such as NIS, LDAP or  SQL,  patterns  are
#        tried in the order as listed below:
#
#        domain transport:nexthop
#               Mail  for  domain is delivered through transport to
#               nexthop.
#
#        .domain transport:nexthop
#               Mail for  any  subdomain  of  domain  is  delivered
#               through  transport  to  nexthop.  This applies only
#               when the string transport_maps is not listed in the
#               parent_domain_matches_subdomains configuration set-
#               ting.  Otherwise, a domain name matches itself  and
#               its subdomains.
#        An empty result (`:' - default transport, default nexthop)
#        behaves as  though the transport  map did not exist.  When
#        combined with a wildcard (`*') entry,  this can be used to
#        route internal mail directly, while  using a relay for all
#        outbound  traffic.    (Note  that  you  should  _NOT_  set
#        relayhost in this case.)
#
#             *                  smtp:outbound-relay.my.domain
```

```
#             .my.domain         :
#
#        Note:  transport  map entries take precedence over domains
#        specified in the mydestination parameter. If you  use  the
#        optional  transport  map,  it  may  be  safer  to  specify
#        explicit entries for all domains specified  in  mydestina-
#        tion, for example:
#
#             hostname.my.domain   local:
#             localhost.my.domain      local:
#
#        The  interpretation  of  the  nexthop  field  is transport
#        dependent. In the case of SMTP, specify host:service for a
#        non-default  server port, and use [host] or [host]:port in
#        order to disable MX (mail exchanger) DNS lookups.  The []
#        form can also be used with IP addresses instead of hostnames.
#
# EXAMPLES
#        In order to send mail for foo.org and its  subdomains  via
#        the uucp transport to the UUCP host named foo:
#
#             foo.org      uucp:foo
#             .foo.org     uucp:foo
#
#        When  no  nexthop  host name is specified, the destination
#        domain name is used instead. For  example,  the  following
#        directs  mail for user@foo.org via the slow transport to a
#        mail exchanger for foo.org.  The slow transport  could  be
#        something  that  runs  at  most  one delivery process at a
#        time:
#
#             foo.org      slow:
#
#        When no transport is specified, the default  transport  is
#        used, as specified via the default_transport configuration
#        parameter. The following sends all mail  for  foo.org  and
#        its subdomains to host gateway.foo.org:
#
#             foo.org      :[gateway.foo.org]
#             .foo.org     :[gateway.foo.org]
#
#        In  the  above  example,  the  []  are used to suppress MX
#        lookups.  The result would  likely  point  to  your  local
#        machine.
#
#        In  the  case  of delivery via SMTP, one may specify host-
#        name:service instead of just a host:
#
#             foo.org      smtp:bar.org:2025
#
#        This directs mail for user@foo.org to  host  bar.org  port
#        2025.  Instead  of a numerical port a symbolic name may be
#        used. Specify [] around the hostname in order  to  disable
#        MX lookups.
#        The error mailer can be used to bounce mail:
#
#             .foo.org       error:mail for *.foo.org is not deliverable
#
#        This causes all mail for user@anything.foo.org to be bounced.
#
# REGULAR EXPRESSION TABLES
#        This  section  describes how the table lookups change when
#        the table is given in the form of regular expressions. For
#        a  description  of regular expression lookup table syntax,
#        see regexp_table(5) or pcre_table(5).
#
#        Each pattern is a regular expression that  is  applied  to
#        the entire domain being looked up. Thus, some.domain.hier-
#        archy is not broken up into parent domains.
#
```

```
#       Patterns are applied in the  order  as  specified  in  the
#       table,  until  a  pattern is found that matches the search
#       string.
#
#       Results are the same as with indexed  file  lookups,  with
#       the  additional feature that parenthesized substrings from
#       the pattern can be interpolated as $1, $2 and so on.
#
# CONFIGURATION PARAMETERS
#       The following main.cf parameters are  especially  relevant
#       to  this  topic.  See  the Postfix main.cf file for syntax
#       details and for default values.  Use  the  postfix  reload
#       command after a configuration change.
#
#       parent_domain_matches_subdomains
#              List  of  Postfix features that use domain.tld pat-
#              terns  to  match  sub.domain.tld  (as  opposed   to
#              requiring .domain.tld patterns).
#
#       transport_maps
#              List of transport lookup tables.
#
#       Other parameters of interest:
#
#       default_transport
#              The  transport  to use when no transport is explic-
#              itly specified.
#
#       relayhost
#              The default host to send to when no transport table
#              entry matches.
#
# SEE ALSO
#       postmap(1) create mapping table
#       trivial-rewrite(8) rewrite and resolve addresses
#       pcre_table(5) format of PCRE tables
#       regexp_table(5) format of POSIX regular expression tables
#
```

```
VIRTUAL(5)                                              VIRTUAL(5)
#
# NAME
#       virtual - format of Postfix virtual table
#
# SYNOPSIS
#       postmap /etc/postfix/virtual
#
# DESCRIPTION
#       The  optional virtual table specifies address redirections
#       for local and non-local recipients or domains.  The  redi-
#       rections  are  used by the cleanup(8) daemon. The redirec-
#       tions are recursive.
#
#       The virtual redirection is applied only to recipient enve-
#       lope  addresses,  and  does  not  affect  message headers.
#       Think Sendmail rule set S0, if you like. Use  canonical(5)
#       mapping  to  rewrite header and envelope addresses in gen-
#       eral.
#
#       Normally, the virtual table is specified as  a  text  file
#       that  serves  as  input  to  the  postmap(1) command.  The
#       result, an indexed file in dbm or db format, is  used  for
#       fast  searching  by  the  mail system. Execute the command
#       postmap  /etc/postfix/virtual  in  order  to  rebuild  the
#       indexed file after changing the text file.
#
#       When  the  table  is provided via other means such as NIS,
#       LDAP or SQL, the same lookups are  done  as  for  ordinary
#       indexed files.
#
#       Alternatively,  the  table  can  be provided as a regular-
#       expression map where patterns are given as regular expres-
#       sions.  In  that  case, the lookups are done in a slightly
#       different way as described below.
#
# POSTFIX-STYLE VIRTUAL DOMAINS
#       With a Postfix-style virtual domain,  the  virtual  domain
#       has  its  own  user  name  space. Local (i.e. non-virtual)
#       usernames are  not  visible  in  a  Postfix-style  virtual
#       domain.  In particular, local aliases(5) and mailing lists
#       are not visible as localname@virtual.domain.
#
#       Use a Sendmail-style virtual domain (see below)  if  local
#       usernames,  aliases(5)  or mailing lists should be visible
#       as localname@virtual.domain.
#
#       Support for a Postfix-style virtual domain looks like:
#
#       /etc/postfix/virtual:
#           virtual.domain       anything (right-hand content does not matter)
#           postmaster@virtual.domain    postmaster
#           user1@virtual.domain address1
#           user2@virtual.domain address2, address3
#
#       The virtual.domain anything entry is required for a  Post-
#       fix-style virtual domain.
#
#       Do  not list a Postfix-style virtual domain in the main.cf
#       mydestination configuration parameter.  Such an  entry  is
#       required only for a Sendmail-style virtual domain.
#
#       With  a  Postfix-style  virtual  domain,  the Postfix SMTP
#       server  accepts  mail  for  known-user@virtual.domain  and
#       rejects mail for unknown-user@virtual.domain as undeliver-
#       able.
#
# SENDMAIL-STYLE VIRTUAL DOMAINS
#       With a Sendmail-style virtual domain,  every  local  (i.e.
#       non-virtual) username is visible in the virtual domain. In
```

```
#       particular, every local alias and mailing list is  visible
#       as localname@virtual.domain.
#
#       Use  a  Postfix-style  virtual domain (see above) if local
#       usernames, aliases(5) or mailing lists should not be visi-
#       ble as localname@virtual.domain.
#
#       Support for a Sendmail-style virtual domain looks like:
#
#       /etc/postfix/main.cf:
#           mydestination = $myhostname localhost.$mydomain $mydomain
#               virtual.domain
#
#       /etc/postfix/virtual:
#           user1@virtual.domain address1
#           user2@virtual.domain address2, address3
#
#       The  main.cf  mydestination  entry is required for a Send-
#       mail-style virtual domain.
#
#       Do not specify a virtual.domain anything virtual map entry
#       for  a  Sendmail-style  virtual  domain.  Such an entry is
#       required only with a Postfix-style virtual domain.
#
#       With a Sendmail-style virtual domain,  the  Postfix  local
#       delivery  agent  delivers  mail  for  an unknown user@vir-
#       tual.domain to a local (i.e.  non-virtual) user  that  has
#       the  same  name;  if no such recipient exists, the Postfix
#       local delivery agent bounces the mail to the sender.
#
# TABLE FORMAT
#       The format of the virtual table is  as  follows,  mappings
#       being tried in the order as listed in this manual page:
#
#       pattern result
#              When  pattern matches a mail address, replace it by
#              the corresponding result.
#
#       blank lines and comments
#              Empty lines and whitespace-only lines are  ignored,
#              as  are  lines whose first non-whitespace character
#              is a `#'.
#
#       multi-line text
#              A logical line starts with non-whitespace  text.  A
#              line  that starts with whitespace continues a logi-
#              cal line.
#
#       With lookups from indexed files such as DB or DBM, or from
#       networked  tables  such  as NIS, LDAP or SQL, patterns are
#       tried in the order as listed below:
#
```

```
#       ***user@domain address, address, ...***
#              Mail for user@domain is redirected to address.
#              This form has the highest precedence.
#
#       ***user address, address, ...***
#              Mail for user@site is redirected to address when
#              site is equal to $myorigin, when site is listed in
#              $mydestination, or when it is listed in
#              $inet_interfaces.
#
#              This functionality overlaps with functionality of
#              the local alias(5) database. The difference is that
#              virtual mapping can be applied to non-local
#              addresses.
#
#       ***@domain address, address, ...***
#              Mail for any user in domain is redirected to
#              address. This form has the lowest precedence.
#
#       In all the above forms, when address has the form @other-
#       domain, the result is the same user in otherdomain. This
#       works for the first address in the expansion only.
#
# **ADDRESS EXTENSION**
#       When a mail address localpart contains the optional recip-
#       ient delimiter (e.g., user+foo@domain), the lookup order
#       becomes: user+foo@domain, user@domain, user+foo, user, and
#       @domain. An unmatched address extension (+foo) is propa-
#       gated to the result of table lookup.
#
# **REGULAR EXPRESSION TABLES**
#       This section describes how the table lookups change when
#       the table is given in the form of regular expressions. For
#       a description of regular expression lookup table syntax,
#       see regexp_table(5) or pcre_table(5).
#
#       Each pattern is a regular expression that is applied to
#       the entire address being looked up. Thus, user@domain mail
#       addresses are not broken up into their user and @domain
#       constituent parts, nor is user+foo broken up into user and
#       foo.
#
#       Patterns are applied in the order as specified in the
#       table, until a pattern is found that matches the search
#       string.
#
#       Results are the same as with indexed file lookups, with
#       the additional feature that parenthesized substrings from
#       the pattern can be interpolated as $1, $2 and so on.
#
# **BUGS**
#       The table format does not understand quoting conventions.

# **CONFIGURATION PARAMETERS**
#       The following main.cf parameters are especially relevant
#       to this topic. See the Postfix main.cf file for syntax
#       details and for default values. Use the postfix reload
#       command after a configuration change.
#
#       **virtual_maps**
#              List of virtual mapping tables.
```

```
#       Other parameters of interest:
#
#       inet_interfaces
#              The  network  interface  addresses that this system
#              receives mail on.
#       mydestination
#              List of domains that  this  mail  system  considers
#              local.
#
#       myorigin
#              The domain that is appended to locally-posted mail.
#
#       owner_request_special
#              Give special treatment to owner-xxx and xxx-request
#              addresses.
#
# SEE ALSO
#       cleanup(8) canonicalize and enqueue mail
#       postmap(1) create mapping table
#       pcre_table(5) format of PCRE tables
#       regexp_table(5) format of POSIX regular expression tables
#
# LICENSE
#       The  Secure  Mailer  license must be distributed with this
#       software.
#
# AUTHOR(S)
#       Wietse Venema
#       IBM T.J. Watson Research
#       P.O. Box 704
#       Yorktown Heights, NY 10598, USA
```

VIRTUAL(5) VIRTUAL(5)

NAME

virtual - format of Postfix virtual alias table

SYNOPSIS

postmap /etc/postfix/virtual

postmap -q "*string*" **/etc/postfix/virtual**

postmap -q - /etc/postfix/virtual <*inputfile*

DESCRIPTION

The optional **virtual** alias table specifies address aliasing for arbitrary local or non-local recipient addresses. Virtual aliasing is recursive, and is done by the Postfix **cleanup**(8) daemon.

The main applications of virtual aliasing are:

- To redirect mail for one address to one or more addresses.

- To implement virtual alias domains where all addresses are aliased to addresses in other domains.

  Virtual alias domains are not to be confused with the virtual mailbox domains that are implemented with the Postfix **virtual**(8) mail delivery agent. With virtual mailbox domains, each recipient address can have its own mailbox.

Virtual aliasing is applied only to recipient envelope addresses, and does not affect message headers. Think Sendmail rule set **S0**, if you like. Use **canonical**(5) mapping to rewrite header and envelope addresses in general.

Normally, the **virtual** alias table is specified as a text file that serves as input to the **postmap**(1) command. The result, an indexed file in **dbm** or **db** format, is used for fast searching by the mail system. Execute the command **postmap /etc/postfix/virtual** in order to rebuild the indexed file after changing the text file.

When the table is provided via other means such as NIS, LDAP or SQL, the same lookups are done as for ordinary indexed files.

Alternatively, the table can be provided as a regular-expression map where patterns are given as regular expressions. In that case, the lookups are done in a slightly different way as described below.

TABLE FORMAT

The format of the virtual table is as follows, mappings being tried in the order as listed in this manual page:

pattern result

When *pattern* matches a mail address, replace it by the corresponding *result*.

blank lines and comments
Empty lines and whitespace-only lines are ignored, as are lines whose first non-whitespace character is a `#'.

multi-line text
A logical line starts with non-whitespace text. A line that starts with whitespace continues a logical line.

With lookups from indexed files such as DB or DBM, or from networked tables such as NIS, LDAP or SQL, patterns are tried in the order as listed below:

user@domain address, address, ...
Mail for *user@domain* is redirected to *address*. This form has the highest precedence.

user address, address, ...
Mail for *user@site* is redirected to *address* when *site* is equal to $**myorigin**, when *site* is listed in $mydestination, or when it is listed in $*inet_interfaces*.

This functionality overlaps with functionality of the local *aliases*(5) database. The difference is that **virtual** mapping can be applied to non-local addresses.

@domain address, address, ...
Mail for any user in *domain* is redirected to *address*. This form has the lowest precedence.

In all the above forms, when *address* has the form *@otherdomain*, the result is the same user in *otherdomain*. This works for the first address in the expansion only.

ADDRESS EXTENSION

When a mail address localpart contains the optional recipient delimiter (e.g., *user+foo@domain*), the lookup order becomes: *user+foo@domain*, *user@domain*, *user+foo*, *user*, and *@domain*. An unmatched address extension (*+foo*) is propagated to the result of table lookup.

VIRTUAL ALIAS DOMAINS

Besides virtual aliases, the virtual alias table can also be used to implement virtual alias domains. With a virtual alias domain, all recipient addresses are aliased to addresses in other domains.

Virtual alias domains are not to be confused with the virtual mailbox domains that are implemented with the Postfix **virtual**(8) mail delivery agent. With virtual mailbox domains, each recipient address can have its own mailbox.

With a virtual alias domain, the virtual domain has its own user name space. Local (i.e. non-virtual) usernames are not visible in a virtual alias domain. In particular, local **aliases**(5) and local mailing lists are not visible as *localname@virtual-alias.domain*.

Support for a virtual alias domain looks like:

```
/etc/postfix/main.cf:
    virtual_alias_maps = hash:/etc/postfix/virtual

    Note: some systems use dbm databases instead of  hash.
    See the output from postconf -m for available database
    types.

/etc/postfix/virtual:
    virtual-alias.domain anything (right-hand content does not matter)
    postmaster@virtual-alias.domain      postmaster
    user1@virtual-alias.domain   address1
    user2@virtual-alias.domain   address2, address3
```

The *virtual-alias.domain anything* entry is required for a virtual alias domain. Without this entry, mail is rejected with "relay access denied", or bounces with "mail loops back to myself".

Do not specify virtual alias domain names in the **main.cf mydestination** or **relay_domains** configuration parameters.

With a virtual alias domain, the Postfix SMTP server accepts mail for *known-user@virtual-alias.domain*, and rejects mail for *unknown-user@virtual-alias.domain* as undeliverable.

Instead of specifying the virtual alias domain name via the **virtual_alias_maps** table, you may also specify it via the **main.cf virtual_alias_domains** configuration parameter. This latter parameter uses the same syntax as the **main.cf mydestination** configuration parameter.

REGULAR EXPRESSION TABLES

This section describes how the table lookups change when the table is given in the form of regular expressions. For a description of regular expression lookup table syntax, see **regexp_table**(5) or **pcre_table**(5).

Each pattern is a regular expression that is applied to the entire address being looked up. Thus, *user@domain* mail addresses are not broken up into their *user* and *@domain* constituent parts, nor is *user+foo* broken up into *user* and *foo*.

Patterns are applied in the order as specified in the table, until a pattern is found that matches the search string.

Results are the same as with indexed file lookups, with the additional feature that parenthesized substrings from the pattern can be interpolated as **$1**, **$2** and so on.

BUGS

The table format does not understand quoting conventions.

CONFIGURATION PARAMETERS

The following **main.cf** parameters are especially relevant to this topic. See the Postfix **main.cf** file for syntax details and for default values. Use the **postfix reload** command after a configuration change.

virtual_alias_maps

List of virtual aliasing tables.

virtual_alias_domains
List of virtual alias domains. This uses the same syntax as the **mydestination** parameter.

Other parameters of interest:

inet_interfaces
The network interface addresses that this system receives mail on. You need to stop and start Postfix when this parameter changes.

mydestination
List of domains that this mail system considers local.

myorigin
The domain that is appended to any address that does not have a domain.

owner_request_special
Give special treatment to **owner-***xxx* and *xxx*-**request** addresses.

SEE ALSO
cleanup(8) canonicalize and enqueue mail
postmap(1) create mapping table
regexp_table(5) POSIX regular expression table format
pcre_table(5) Perl Compatible Regular Expression table format

LICENSE
The Secure Mailer license must be distributed with this software.

AUTHOR(S)
Wietse Venema
IBM T.J. Watson Research
P.O. Box 704
Yorktown Heights, NY 10598, USA

VIRTUAL(5)

```
# REGEXP_TABLE(5)                                  REGEXP_TABLE(5)

#
# NAME
#       regexp_table - format of Postfix regular expression tables
#
# SYNOPSIS
#       regexp:/etc/postfix/filename
#
# DESCRIPTION
#       The Postfix mail system uses optional tables  for  address
#       rewriting or mail routing. These tables are usually in dbm
#       or db format. Alternatively, lookup tables can  be  speci-
#       fied in POSIX regular expression form.
#
#       To  find out what types of lookup tables your Postfix sys-
#       tem supports use the postconf -m command.
#
#       The general form of a Postfix regular expression table is:
#
#       pattern result
#              When  pattern matches a search string, use the cor-
#              responding result.
#
#       blank lines and comments
#              Empty lines and whitespace-only lines are  ignored,
#              as  are  lines whose first non-whitespace character
#              is a `#'.
#
#       multi-line text
#              A logical line starts with non-whitespace  text.  A
#              line  that starts with whitespace continues a logi-
#              cal line.
#
#       pattern1!pattern2 result
#              Matches pattern1 but not pattern2.
#
#       Each pattern is a regular expression enclosed by a pair of
#       delimiters.  The regular expression syntax is described in
#       re_format(7).  The expression delimiter can be any charac-
#       ter,  except  whitespace  or  characters that have special
#       meaning (traditionally the forward  slash  is  used).  The
#       regular expression can contain whitespace.
#
#       By default, matching is case-insensitive, although follow-
#       ing the second slash with an `i' flag will  reverse  this.
#       Other  flags are `x' (disable extended expression syntax),
#       and `m' (enable multi-line mode).
#
#       Each pattern is applied to the entire lookup  key  string.
#       Depending  on  the  application,  that string is an entire
#       client hostname, an entire client IP address, or an entire
#       mail  address.   Thus,  no parent domain or parent network
#       search is done, and user@domain  mail  addresses  are  not
#       broken  up  into  their user and domain constituent parts,
#       nor is user+foo broken up into user and foo.
#
#       Patterns are applied in the  order  as  specified  in  the
#       table,  until  a  pattern is found that matches the search
#       string.
#
#       Substitution of substrings  from  the  matched  expression
#       into the result string is possible using $1, $2, etc.. The
#       macros in the result string may need to be written as ${n}
#       or $(n) if they aren't followed by whitespace.
# EXAMPLE SMTPD ACCESS MAP
#       # Disallow sender-specified routing. This is a must if you relay mail
#       # for other domains.
#       /[%!@].*[%!@]/       550 Sender-specified routing rejected
```

```
#
#       # Postmaster is OK, that way they can talk to us about how to fix
#       # their problem.
#       /^postmaster@/       OK
#
#       # Protect your outgoing majordomo exploders
#       /^(.*)-outgoing@(.*)$/!/^owner-/         550 Use ${1}@${2} instead
#
# EXAMPLE HEADER FILTER MAP
#       # These were once common in junk mail.
#       /^Subject: make money fast/     REJECT
#       /^To: friend@public\.com/       REJECT
#
# SEE ALSO
#       pcre_table(5) format of PCRE tables
#
# AUTHOR(S)
#       The regexp table lookup code was originally written by:
#       LaMont Jones
#       lamont@hp.com
#
#       That code was based on the PCRE dictionary contributed by:
#       Andrew McNamara
#       andrewm@connect.com.au
#       connect.com.au Pty. Ltd.
#       Level 3, 213 Miller St
#       North Sydney, NSW, Australia
#
#       Adopted and adapted by:
#       Wietse Venema
#       IBM T.J. Watson Research
#       P.O. Box 704
#       Yorktown Heights, NY 10598, USA
```

/etc/postfix/dynamicmaps.cf

```
# Postfix dynamic maps configuration file.
#
# The first match found is the one that is used.  The only wildcard
# allowed is '*', which matches everything.  The first %s is expanded
# to the map type.
#
#type location of .so file              name of open function
#==== ================================  ====================
/usr/lib/postfix/dict_%s.so             dict_%s_open
```

Programs running under Postfix

Postfix background processes

The previous sections gave a simplified overview of how the Postfix system sends and receives mail. Several other things happen behind the scenes. Unfortunately, this is hard to visualize on a two-dimensional display, so this document has no illustration.

- The master daemon is the supervisor process that keeps an eye on the well-being of the mail system. It is typically started at system boot time by the postfix command, and keeps running until the system goes down. The master daemon is responsible for starting all other Postfix daemon processes on demand, and for restarting daemons that terminated prematurely because of some problem. The master daemon is also responsible for enforcing the daemon process count limits as specified in the **master.cf** configuration file.
- The bounce or defer daemon is called upon left and right by other daemon processes, in order to maintain per-message log files with non-delivery status information.
- The trivial-rewrite daemon is called upon left and right by other daemon processes, in order to rewrite an address to *user@fully.qualified.domain* form, or in order to resolve a destination.
- The showq daemon lists the Postfix queue status. This is the program behind the mailq command.
- The flush daemon improves the performance of the SMTP **ETRN** request, and of its command-line equivalent, **sendmail -qR***destination*, for selected destinations.
- The proxymap daemon provides read-only lookup service to Postfix client processes. The purpose is to overcome chroot restrictions, and to consolidate the number of open lookup tables by sharing one open table among multiple processes.
- The spawn daemon listens on a TCP port, UNIX-domain socket or FIFO, and runs non-Postfix commands on request, with the socket or FIFO connected to the standard input, output and error streams. It is currently used only in an example of the Postfix external content filtering system.

BOUNCE(8) BOUNCE(8)

NAME

bounce - Postfix message bounce or defer daemon

SYNOPSIS

bounce [generic Postfix daemon options]

DESCRIPTION

The **bounce** daemon maintains per-message log files with non-delivery status information. Each log file is named after the queue file that it corresponds to, and is kept in a queue subdirectory named after the service name in the **master.cf** file (either **bounce** or **defer**). This program expects to be run from the **master**(8) process manager.

The **bounce** daemon processes two types of service requests:

- Append a recipient status record to a per-message log file.
- Post a bounce message, with a copy of a log file and of the corresponding message. When the bounce is posted successfully, the log file is deleted.

The software does a best effort to notify the sender that there was a problem. A notification is sent even when the log file or original message cannot be read.

Optionally, a client can request that the per-message log file be deleted when the requested operation fails. This is used by clients that cannot retry transactions by themselves, and that depend on retry logic in their own client.

STANDARDS

RFC 822 (ARPA Internet Text Messages)
RFC 1894 (Delivery Status Notifications)
RFC 2045 (Format of Internet Message Bodies)

DIAGNOSTICS

Problems and transactions are logged to **syslogd**(8).

BUGS

The log files use an ad-hoc, unstructured format. This will have to change in order to easily support standard delivery status notifications.

CONFIGURATION PARAMETERS

The following **main.cf** parameters are especially relevant to this program. See the Postfix **main.cf** file for syntax details and for default values. Use the **postfix reload** command after a configuration change.

bounce_notice_recipient
The recipient of single bounce postmaster notices.

2bounce_notice_recipient
The recipient of double bounce postmaster notices.

delay_notice_recipient
The recipient of "delayed mail" postmaster notices.

bounce_size_limit

Limit the amount of original message context that is sent in a non-delivery notification.

mail_name
Use this mail system name in the introductory text at the start of a bounce message.

notify_classes
Notify the postmaster of bounced mail when this parameter includes the **bounce** class. For privacy reasons, the message body is not included.

SEE ALSO

master(8) process manager
qmgr(8) queue manager
syslogd(8) system logging

LICENSE

The Secure Mailer license must be distributed with this software.

AUTHOR(S)

Wietse Venema
IBM T.J. Watson Research
P.O. Box 704
Yorktown Heights, NY 10598, USA

BOUNCE(8)

MASTER(8) MASTER(8)

NAME

master - Postfix master process

SYNOPSIS

master [**-Dtv**] [**-c** *config_dir*] [**-e** *exit_time*]

DESCRIPTION

The **master** daemon is the resident process that runs Postfix daemons on demand: daemons to send or receive messages via the network, daemons to deliver mail locally, etc. These daemons are created on demand up to a configurable maximum number per service.

Postfix daemons terminate voluntarily, either after being idle for a configurable amount of time, or after having serviced a configurable number of requests. The exception to this rule is the resident Postfix queue manager.

The behavior of the **master** daemon is controlled by the **master.cf** configuration file. The table specifies zero or more servers in the **UNIX** or **INET** domain, or servers that take requests from a FIFO. Precise configuration details are given in the **master.cf** file, and in the manual pages of the respective daemons.

Options:

-c *config_dir*
Read the **main.cf** and **master.cf** configuration files in the named directory instead of the default configuration directory.

-e *exit_time*
Terminate the master process after *exit_time* seconds. Child processes terminate at their convenience.

-D After initialization, run a debugger on the master process. The debugging command is specified with the **debugger_command** in the **main.cf** global configuration file.

-t Test mode. Return a zero exit status when the **master.pid** lock file does not exist or when that file is not locked. This is evidence that the **master** daemon is not running.

-v Enable verbose logging for debugging purposes. This option is passed on to child processes. Multiple **-v** options make the software increasingly verbose.

Signals:

SIGHUP Upon receipt of a **HUP** signal (e.g., after **postfix reload**), the master process re-reads its configuration files. If a service has been removed from the **master.cf** file, its running processes are terminated immediately. Otherwise, running processes are allowed to terminate as soon as is convenient, so that changes in configuration settings affect

only new service requests.

SIGTERM
Upon receipt of a **TERM** signal (e.g., after **postfix abort**), the master process passes the signal on to its child processes and terminates. This is useful for an emergency shutdown. Normally one would terminate only the master (**postfix stop**) and allow running processes to finish what they are doing.

DIAGNOSTICS
Problems are reported to **syslogd**(8).

BUGS

ENVIRONMENT

MAIL_DEBUG
After initialization, start a debugger as specified with the **debugger_command** configuration parameter in the **main.cf** configuration file.

MAIL_CONFIG
Directory with Postfix configuration files.

CONFIGURATION PARAMETERS
The following **main.cf** parameters are especially relevant to this program. See the Postfix **main.cf** file for syntax details and for default values. Use the **postfix reload** command after a configuration change.

Miscellaneous

import_environment

export_environment
Lists of names of environment parameters that can be imported from (exported to) non-Postfix processes.

mail_owner
The owner of the mail queue and of most Postfix processes.

command_directory
Directory with Postfix support programs.

daemon_directory
Directory with Postfix daemon programs.

queue_directory
Top-level directory of the Postfix queue. This is also the root directory of Postfix daemons that run chrooted.

inet_interfaces
The network interface addresses that this system receives mail on. You need to stop and start Postfix when this parameter changes.

Resource controls

default_process_limit
Default limit for the number of simultaneous child processes that provide a given service.

max_idle
Limit the time in seconds that a child process waits between service requests.

max_use
Limit the number of service requests handled by a child process.

service_throttle_time
Time to avoid forking a server that appears to be broken.

FILES

/etc/postfix/main.cf: global configuration file.
/etc/postfix/master.cf: master process configuration file.
/var/spool/postfix/pid/master.pid: master lock file.

SEE ALSO

qmgr(8) queue manager
pickup(8) local mail pickup
syslogd(8) system logging

LICENSE

The Secure Mailer license must be distributed with this software.

AUTHOR(S)

Wietse Venema
IBM T.J. Watson Research
P.O. Box 704
Yorktown Heights, NY 10598, USA

MASTER(8)

TRIVIAL-REWRITE(8) TRIVIAL-REWRITE(8)

NAME

trivial-rewrite - Postfix address rewriting and resolving daemon

SYNOPSIS

trivial-rewrite [generic Postfix daemon options]

DESCRIPTION

The **trivial-rewrite** daemon processes two types of client service requests:

rewrite

Rewrite an address to standard form. The **trivial-rewrite** daemon by default appends local domain information to unqualified addresses, swaps bang paths to domain form, and strips source routing information. This process is under control of several configuration parameters (see below).

resolve

Resolve an address to a (*transport*, *nexthop*, *recipient*) triple. The meaning of the results is as follows:

transport

The delivery agent to use. This is the first field of an entry in the **master.cf** file.

nexthop

The host to send to and optional delivery method information.

recipient

The envelope recipient address that is passed on to *nexthop*.

DEFAULT DELIVERY METHODS

By default, Postfix uses one of the following delivery methods. This may be overruled with the optional transport(5) table. The default delivery method is selected by matching the recipient address domain against one of the following:

$mydestination

$inet_interfaces

The transport and optional nexthop are specified with **$local_transport**. The default nexthop is the recipient domain.

$virtual_alias_domains

The recipient address is undeliverable (user unknown). By definition, all known addresses in a virtual alias domain are aliased to other addresses.

$virtual_mailbox_domains

The transport and optional nexthop are specified with **$virtual_transport**. The default nexthop is the recipient domain.

$relay_domains

The transport and optional nexthop are specified with **$relay_transport**. This overrides the optional

nexthop information that is specified with **$relay-host**. The default nexthop is the recipient domain.

none of the above
The transport and optional nexthop are specified with **$default_transport**. This overrides the optional nexthop information that is specified with **$relayhost**. The default nexthop is the recipient domain.

SERVER PROCESS MANAGEMENT
The trivial-rewrite servers run under control by the Postfix master server. Each server can handle multiple simultaneous connections. When all servers are busy while a client connects, the master creates a new server process, provided that the trivial-rewrite server process limit is not exceeded. Each trivial-rewrite server terminates after serving at least **$max_use** clients of after **$max_idle** seconds of idle time.

STANDARDS
None. The command does not interact with the outside world.

SECURITY
The **trivial-rewrite** daemon is not security sensitive. By default, this daemon does not talk to remote or local users. It can run at a fixed low privilege in a chrooted environment.

DIAGNOSTICS
Problems and transactions are logged to **syslogd**(8).

BUGS
CONFIGURATION PARAMETERS
The following **main.cf** parameters are especially relevant to this program. See the Postfix **main.cf** file for syntax details and for default values. Use the **postfix reload** command after a configuration change.

Miscellaneous
empty_address_recipient
The recipient that is substituted for the null address.

inet_interfaces
The network interfaces that this mail system receives mail on. This information is used to determine if *user*@[*net.work.addr.ess*] is local or remote. Mail for local users is given to the **$local_transport**.

mydestination
List of domains that are given to the **$local_transport**.

virtual_alias_domains
List of simulated virtual domains (domains with all recipients aliased to some other local or remote domain).

virtual_mailbox_domains
List of domains that are given to the **$virtual_transport**.

relay_domains
List of domains that are given to the **$relay_transport**.

resolve_unquoted_address
When resolving an address, do not quote the address localpart as per RFC 822, so that additional **@**, **%** or **!** characters remain visible. This is technically incorrect, but allows us to stop relay attacks when forwarding mail to a Sendmail primary MX host.

relocated_maps
Tables with contact information for users, hosts or domains that no longer exist. See **relocated**(5).

Rewriting

myorigin
The domain that locally-posted mail appears to come from.

allow_percent_hack
Rewrite *user%domain* to *user@domain*.

append_at_myorigin
Rewrite *user* to *user*@**$myorigin**.

append_dot_mydomain
Rewrite *user@host* to *user@host*.**$mydomain**.

swap_bangpath
Rewrite *site*!*user* to *user@site*.

Routing

local_transport
Where to deliver mail for destinations that match **$mydestination** or **$inet_interfaces**. The default transport is **local:$myhostname**.

Syntax is *transport*:*nexthop*; see **transport**(5) for details. The :*nexthop* part is optional.

virtual_transport
Where to deliver mail for non-local domains that match **$virtual_mailbox_domains**. The default transport is **virtual**.

Syntax is *transport*:*nexthop*; see **transport**(5) for details. The :*nexthop* part is optional.

relay_transport
Where to deliver mail for non-local domains that match **$relay_domains**. The default transport is **relay** (which normally is a clone of the **smtp** transport).

Syntax is *transport*:*nexthop*; see **transport**(5) for details. The :*nexthop* part is optional.

default_transport
Where to deliver all other non-local mail. The default transport is **smtp**.

Syntax is *transport*:*nexthop*; see **transport**(5) for

details. The :*nexthop* part is optional.

parent_domain_matches_subdomains
List of Postfix features that use *domain.tld* patterns to match *sub.domain.tld* (as opposed to requiring *.domain.tld* patterns).

relayhost
The default host to send non-local mail to when no host is specified with **$relay_transport** or **$default_transport**, and when the recipient address does not match the optional the **transport**(5) table.

transport_maps
List of tables with *recipient* or *domain* to (*transport, nexthop*) mappings.

SEE ALSO

master(8) process manager
syslogd(8) system logging
transport(5) transport table format
relocated(5) format of the "user has moved" table

LICENSE

The Secure Mailer license must be distributed with this software.

AUTHOR(S)

Wietse Venema
IBM T.J. Watson Research
P.O. Box 704
Yorktown Heights, NY 10598, USA

TRIVIAL-REWRITE(8)

SHOWQ(8) SHOWQ(8)

NAME

showq - list the Postfix mail queue

SYNOPSIS

showq [generic Postfix daemon options]

DESCRIPTION

The **showq** daemon reports the Postfix mail queue status. It is the program that emulates the sendmail `mailq' command.

The **showq** daemon can also be run in stand-alone mode by the superuser. This mode of operation is used to emulate the `mailq' command while the Postfix mail system is down.

SECURITY

The **showq** daemon can run in a chroot jail at fixed low privilege, and takes no input from the client. Its service port is accessible to local untrusted users, so the service can be susceptible to denial of service attacks.

STANDARDS

None. The showq daemon does not interact with the outside world.

DIAGNOSTICS

Problems and transactions are logged to **syslogd**(8).

BUGS

The **showq** daemon runs at a fixed low privilege; consequently, it cannot extract information from queue files in the **maildrop** directory.

SEE ALSO

cleanup(8) canonicalize and enqueue mail
pickup(8) local mail pickup service
qmgr(8) mail being delivered, delayed mail
syslogd(8) system logging

LICENSE

The Secure Mailer license must be distributed with this software.

AUTHOR(S)

Wietse Venema
IBM T.J. Watson Research
P.O. Box 704
Yorktown Heights, NY 10598, USA

SHOWQ(8)

FLUSH(8) FLUSH(8)

NAME

flush - Postfix fast flush server

SYNOPSIS

flush [generic Postfix daemon options]

DESCRIPTION

The flush server maintains a record of deferred mail by destination. This information is used to improve the performance of the SMTP **ETRN** request, and of its command-line equivalent, **sendmail -qR**. This program expects to be run from the **master**(8) process manager.

The record is implemented as a per-destination logfile with as contents the queue IDs of deferred mail. A logfile is append-only, and is truncated when delivery is requested for the corresponding destination. A destination is the part on the right-hand side of the right-most **@** in an email address.

Per-destination logfiles of deferred mail are maintained only for eligible destinations. The list of eligible destinations is specified with the **fast_flush_domains** configuration parameter, which defaults to **$relay_domains**.

This server implements the following requests:

FLUSH_REQ_ADD *sitename queue_id*

Inform the fast flush server that the specified message is queued for *sitename*. Depending on logging policy, the fast flush server stores or ignores the information.

FLUSH_REQ_SEND *sitename*

Request delivery of mail that is queued for *sitename*. If the destination is eligible for a fast flush logfile, this request triggers delivery of messages listed in that destination's logfile, and the logfile is truncated to zero length; if mail is undeliverable it will be added back to the logfile.

If the destination is not eligible for a fast flush logfile, this request is rejected (see below for status codes).

TRIGGER_REQ_WAKEUP

This wakeup request from the master is an alternative way to request **FLUSH_REQ_REFRESH**.

FLUSH_REQ_REFRESH (completes in the background)

Refresh non-empty per-destination logfiles that were not read in **fast_flush_refresh_time** hours, by simulating send requests (see above) for the corresponding destinations.

Delete empty per-destination logfiles that were not updated in **fast_flush_purge_time** days.

FLUSH_REQ_PURGE (completes in the background)

Refresh all non-empty per-destination logfiles, by simulating send requests (see above) for the corresponding destinations. This can be incredibly

expensive when logging is enabled for many destinations, and is not recommended.
Delete empty per-destination logfiles that were not updated in **fast_flush_purge_time** days.

The server response is one of:

FLUSH_STAT_OK
The request completed normally.

FLUSH_STAT_BAD
The flush server rejected the request (bad request name, bad request parameter value).

FLUSH_STAT_FAIL
The request failed.

FLUSH_STAT_DENY
The request was denied because the destination domain is not eligible for fast flush service, or because the fast flush service is disabled.

SECURITY

The fast flush server is not security-sensitive. It does not talk to the network, and it does not talk to local users. The fast flush server can run chrooted at fixed low privilege.

DIAGNOSTICS

Problems and transactions are logged to **syslogd**(8).

BUGS

Fast flush logfiles are truncated only after a **FLUSH_REQ_SEND** request, not when mail is actually delivered, and therefore can accumulate outdated or redundant data. In order to maintain sanity, **FLUSH_REQ_REFRESH** must be executed periodically. This can be automated with a suitable wakeup timer setting in the **master.cf** configuration file.

Upon receipt of a request to deliver all mail for an eligible destination, the **flush** server requests delivery of all messages that are listed in that destination's logfile, regardless of the recipients of those messages. This is not an issue for mail that is sent to a **relay_domains** destination because such mail typically only has recipients in one domain.

FILES

/var/spool/postfix/flush, location of "fast flush" logfiles.

CONFIGURATION PARAMETERS

See the Postfix **main.cf** file for syntax details and for default values. Use the **postfix reload** command after a configuration change.

fast_flush_domains
What destinations can have a "fast flush" logfile. By default, this is set to **$relay_domains**.

fast_flush_refresh_time
Refresh a non-empty "fast flush" logfile that was

```
not read in this amount of time (default time unit:
hours), by simulating a send request for the corre-
sponding destination.
```

fast_flush_purge_time
Remove an empty "fast flush" logfile that was not updated in this amount of time (default time unit: days).

parent_domain_matches_subdomains
List of Postfix features that use *domain.tld* patterns to match *sub.domain.tld* (as opposed to requiring *.domain.tld* patterns).

SEE ALSO
smtpd(8) Postfix SMTP server
qmgr(8) Postfix queue manager
syslogd(8) system logging

LICENSE
The Secure Mailer license must be distributed with this software.

AUTHOR(S)
Wietse Venema
IBM T.J. Watson Research
P.O. Box 704
Yorktown Heights, NY 10598, USA

FLUSH(8)

SENDMAIL(1) SENDMAIL(1)

NAME

sendmail - Postfix to Sendmail compatibility interface

SYNOPSIS

sendmail [*option ...*] [*recipient ...*]

mailq
sendmail -bp

newaliases
sendmail -I

DESCRIPTION

The **sendmail** program implements the Postfix to Sendmail compatibility interface. For the sake of compatibility with existing applications, some Sendmail command-line options are recognized but silently ignored.

By default, **sendmail** reads a message from standard input until EOF or until it reads a line with only a . character, and arranges for delivery. **sendmail** relies on the **postdrop**(1) command to create a queue file in the **maildrop** directory.

Specific command aliases are provided for other common modes of operation:

mailq List the mail queue. Each entry shows the queue file ID, message size, arrival time, sender, and the recipients that still need to be delivered. If mail could not be delivered upon the last attempt, the reason for failure is shown. This mode of operation is implemented by executing the **postqueue**(1) command.

newaliases
Initialize the alias database. If no input file is specified (with the **-oA** option, see below), the program processes the file(s) specified with the **alias_database** configuration parameter. If no alias database type is specified, the program uses the type specified with the **default_database_type** configuration parameter. This mode of operation is implemented by running the **postalias**(1) command.

Note: it may take a minute or so before an alias database update becomes visible. Use the **postfix reload** command to eliminate this delay.

These and other features can be selected by specifying the appropriate combination of command-line options. Some features are controlled by parameters in the **main.cf** configuration file.

The following options are recognized:

-Am (ignored)

-Ac (ignored)
Postfix sendmail uses the same configuration file

regardless of whether or not a message is an initial submission.

-B *body_type*
The message body MIME type: **7BIT** or **8BITMIME**.

-C *config_file* (ignored :-)
The path name of the **sendmail.cf** file. Postfix configuration files are kept in **/etc/postfix**.

-F *full_name*
Set the sender full name. This is used only with messages that have no **From:** message header.

-G (ignored)
Gateway (relay) submission, as opposed to initial user submission.

-I Initialize alias database. See the **newaliases** command above.

-L *label* (ignored)
The logging label. Use the **syslog_name** configuration parameter instead.

-N *dsn* (ignored)
Delivery status notification control. Currently, Postfix does not implement **DSN**.

-R *return_limit* (ignored)
Limit the size of bounced mail. Use the **bounce_size_limit** configuration parameter instead.

-X *log_file* (ignored)
Log mailer traffic. Use the **debug_peer_list** and **debug_peer_level** configuration parameters instead.

-U (ignored)
Initial user submission.

-V Variable Envelope Return Path. Given an envelope sender address of the form *owner-listname*@*origin*, each recipient *user*@*domain* receives mail with a personalized envelope sender address.

By default, the personalized envelope sender address is *owner-listname*+*user*=*domain*@*origin*. The default **+** and **=** characters are configurable with the **default_verp_delimiters** configuration parameter.

-V*xy* As **-V**, but uses *x* and *y* as the VERP delimiter characters, instead of the characters specified with the **default_verp_delimiters** configuration parameter.

-bd Go into daemon mode. This mode of operation is implemented by executing the **postfix start** command.

-bi Initialize alias database. See the **newaliases** command above.

-bm Read mail from standard input and arrange for

delivery. This is the default mode of operation.

-bp List the mail queue. See the **mailq** command above.

-bs Stand-alone SMTP server mode. Read SMTP commands from standard input, and write responses to standard output. In stand-alone SMTP server mode, UCE restrictions and access controls are disabled by default. To enable them, run the process as the **mail_owner** user.

This mode of operation is implemented by running the **smtpd**(8) daemon.

-f *sender*
Set the envelope sender address. This is the address where delivery problems are sent to, unless the message contains an **Errors-To:** message header.

-h *hop_count* (ignored)
Hop count limit. Use the **hopcount_limit** configuration parameter instead.

-i When reading a message from standard input, don't treat a line with only a . character as the end of input.

-m (ignored)
Backwards compatibility.

-n (ignored)
Backwards compatibility.

-oA*alias_database*
Non-default alias database. Specify *pathname* or *type*:*pathname*. See **postalias**(1) for details.

-o7 (ignored)

-o8 (ignored)
To send 8-bit or binary content, use an appropriate MIME encapsulation and specify the appropriate **-B** command-line option.

-oi When reading a message from standard input, don't treat a line with only a . character as the end of input.

-om (ignored)
The sender is never eliminated from alias etc. expansions.

-o *x value* (ignored)
Set option *x* to *value*. Use the equivalent configuration parameter in **main.cf** instead.

-r *sender*
Set the envelope sender address. This is the address where delivery problems are sent to, unless the message contains an **Errors-To:** message header.

-q Attempt to deliver all queued mail. This is imple-

mented by executing the **postqueue**(1) command.

-q*interval* (ignored)
The interval between queue runs. Use the **queue_run_delay** configuration parameter instead.

-qR*site*
Schedule immediate delivery of all mail that is queued for the named *site*. This option accepts only *site* names that are eligible for the "fast flush" service, and is implemented by executing the **postqueue**(1) command. See **flush**(8) for more information about the "fast flush" service.

-qS*site*
This command is not implemented. Use the slower **sendmail -q** command instead.

-t Extract recipients from message headers. This requires that no recipients be specified on the command line.

-v Enable verbose logging for debugging purposes. Multiple **-v** options make the software increasingly verbose. For compatibility with mailx and other mail submission software, a single **-v** option produces no output.

SECURITY

By design, this program is not set-user (or group) id. However, it must handle data from untrusted users or untrusted machines. Thus, the usual precautions need to be taken against malicious inputs.

DIAGNOSTICS

Problems are logged to **syslogd**(8) and to the standard error stream.

ENVIRONMENT

MAIL_CONFIG
Directory with Postfix configuration files.

MAIL_VERBOSE
Enable verbose logging for debugging purposes.

MAIL_DEBUG
Enable debugging with an external command, as specified with the **debugger_command** configuration parameter.

FILES

/var/spool/postfix, mail queue
/etc/postfix, configuration files

CONFIGURATION PARAMETERS

See the Postfix **main.cf** file for syntax details and for default values. Use the **postfix reload** command after a configuration change.

alias_database
Default alias database(s) for **newaliases**. The default value for this parameter is system-specific.

bounce_size_limit
The amount of original message context that is sent along with a non-delivery notification.

default_database_type
Default alias etc. database type. On many UNIX systems the default type is either **dbm** or **hash**.

debugger_command
Command that is executed after a Postfix daemon has initialized.

debug_peer_level
Increment in verbose logging level when a remote host matches a pattern in the **debug_peer_list** parameter.

debug_peer_list
List of domain or network patterns. When a remote host matches a pattern, increase the verbose logging level by the amount specified in the **debug_peer_level** parameter.

default_verp_delimiters
The VERP delimiter characters that are used when the **-V** command line option is specified without delimiter characters.

fast_flush_domains
List of domains that will receive "fast flush" service (default: all domains that this system is willing to relay mail to). This list specifies the domains that Postfix accepts in the SMTP **ETRN** request and in the **sendmail -qR** command.

fork_attempts
Number of attempts to **fork**() a process before giving up.

fork_delay
Delay in seconds between successive **fork**() attempts.

hopcount_limit
Limit the number of **Received:** message headers.

mail_owner
The owner of the mail queue and of most Postfix processes.

command_directory
Directory with Postfix support commands.

daemon_directory
Directory with Postfix daemon programs.

queue_directory
Top-level directory of the Postfix queue. This is also the root directory of Postfix daemons that run chrooted.

queue_run_delay

The time between successive scans of the deferred queue.

verp_delimiter_filter
The characters that Postfix accepts as VERP delimiter characters.

PROXYMAP(8) PROXYMAP(8)

NAME

proxymap - Postfix lookup table proxy server

SYNOPSIS

proxymap [generic Postfix daemon options]

DESCRIPTION

The **proxymap** server provides read-only table lookup service to Postfix client processes. The purpose of the service is:

- To overcome chroot restrictions. For example, a chrooted SMTP server needs access to the system passwd file in order to reject mail for non-existent local addresses, but it is not practical to maintain a copy of the passwd file in the chroot jail. The solution:

  ```
  local_recipient_maps =
      proxy:unix:passwd.byname $alias_maps
  ```

- To consolidate the number of open lookup tables by sharing one open table among multiple processes. For example, making mysql connections from every Postfix daemon process results in "too many connections" errors. The solution:

  ```
  virtual_alias_maps =
      proxy:mysql:/etc/postfix/virtual_alias.cf
  ```

  The total number of connections is limited by the number of proxymap server processes.

The proxymap server implements the following requests:

PROXY_REQ_OPEN *maptype:mapname flags*
Open the table with type *maptype* and name *mapname*, as controlled by *flags*. The reply is the request completion status code (below) and the map type dependent flags.

PROXY_REQ_LOOKUP *maptype:mapname flags key*
Look up the data stored under the requested key. The reply is the request completion status code (below) and the lookup result value. The *maptype:mapname* and *flags* are the same as with the **PROXY_REQ_OPEN** request.

There is no close command, nor are tables implicitly closed when a client disconnects. One of the purposes of the proxymap server is to share tables among multiple client processes.

The request completion status code is one of:

PROXY_STAT_OK
The specified table was opened, or the requested entry was found.

PROXY_STAT_NOKEY

```
The requested table entry was not found.
```

PROXY_STAT_BAD
The request was rejected (bad request parameter value).

PROXY_STAT_RETRY
The lookup request could not be completed.

PROXY_STAT_DENY
The specified table was not approved for access via the proxymap service.

SERVER PROCESS MANAGEMENT
The proxymap servers run under control by the Postfix master server. Each server can handle multiple simultaneous connections. When all servers are busy while a client connects, the master creates a new proxymap server process, provided that the proxymap server process limit is not exceeded. Each proxymap server terminates after serving at least **$max_use** clients or after **$max_idle** seconds of idle time.

SECURITY
The proxymap server opens only tables that are approved via the **proxy_read_maps** configuration parameter, does not talk to users, and can run at fixed low privilege, chrooted or not. However, running the proxymap server chrooted severely limits usability, because it can open only chrooted tables.

The proxymap server is not a trusted daemon process, and must not be used to look up sensitive information such as user or group IDs, mailbox file/directory names or external commands.

DIAGNOSTICS
Problems and transactions are logged to **syslogd**(8).

BUGS
The proxymap server provides service to multiple clients, and must therefore not be used for tables that have high-latency lookups.

CONFIGURATION PARAMETERS
The following main.cf parameters are especially relevant to this program. Use the **postfix reload** command after a configuration change.

proxy_read_maps
A list of zero or more parameter values that may contain references to Postfix lookup tables. Only table references that begin with **proxy:** are approved for read-only access via the proxymap server.

SEE ALSO
dict_proxy(3) proxy map client

LICENSE
The Secure Mailer license must be distributed with this software.

AUTHOR(S)
Wietse Venema
IBM T.J. Watson Research

P.O. Box 704
Yorktown Heights, NY 10598, USA

NAME

spawn - Postfix external command spawner

SYNOPSIS

spawn [generic Postfix daemon options] command_attributes...

DESCRIPTION

The **spawn** daemon provides the Postfix equivalent of **inetd**. It listens on a port as specified in the Postfix **master.cf** file and spawns an external command whenever a connection is established. The connection can be made over local IPC (such as UNIX-domain sockets) or over non-local IPC (such as TCP sockets). The command's standard input, output and error streams are connected directly to the communication endpoint.

This daemon expects to be run from the **master**(8) process manager.

COMMAND ATTRIBUTE SYNTAX

The external command attributes are given in the **master.cf** file at the end of a service definition. The syntax is as follows:

user=*username* (required)

user=*username*:*groupname*

The external command is executed with the rights of the specified *username*. The software refuses to execute commands with root privileges, or with the privileges of the mail system owner. If *groupname* is specified, the corresponding group ID is used instead of the group ID of of *username*.

argv=*command*... (required)

The command to be executed. This must be specified as the last command attribute. The command is executed directly, i.e. without interpretation of shell meta characters by a shell command interpreter.

BUGS

In order to enforce standard Postfix process resource controls, the **spawn** daemon runs only one external command at a time. As such, it presents a noticeable overhead by wasting precious process resources. The **spawn** daemon is expected to be replaced by a more structural solution.

DIAGNOSTICS

The **spawn** daemon reports abnormal child exits. Problems are logged to **syslogd**(8).

SECURITY

This program needs root privilege in order to execute external commands as the specified user. It is therefore security sensitive. However the **spawn** daemon does not talk to the external command and thus is not vulnerable to data-driven attacks.

CONFIGURATION PARAMETERS

The following **main.cf** parameters are especially relevant to this program. See the Postfix **main.cf** file for syntax details and for default values. Use the **postfix reload** command after a configuration change.

Miscellaneous

export_environment

List of names of environment parameters that can be exported to non-Postfix processes.

mail_owner

The process privileges used while not running an external command.

Resource control

*service*_**command_time_limit**

The amount of time the command is allowed to run before it is killed with force. The *service* name is the name of the entry in the **master.cf** file. The default time limit is given by the global **command_time_limit** configuration parameter.

SEE ALSO

master(8) process manager
syslogd(8) system logging

LICENSE

The Secure Mailer license must be distributed with this software.

AUTHOR(S)

Wietse Venema
IBM T.J. Watson Research
P.O. Box 704
Yorktown Heights, NY 10598, USA

1

Postfix tools

Enough daemon talk. The anatomy lesson ends with an introduction to command-line utilities for day-to-day use of the Postfix mail system. Besides the sendmail, mailq, and newaliases commands that were already introduced, the Postfix system comes with it own collection of utilities. For consistency, these are all named post*something*.

- The postfix command controls the operation of the mail system. It is the interface for starting and stopping the mail system, and for some other administrative operations. This command is reserved to the super-user.
- The postalias command maintains Postfix alias databases. This is the program behind the newaliases command.
- The postcat command displays the contents of Postfix queue files. This is a limited, preliminary utility. This program is likely to be superseded by something more powerful that can also edit Postfix queue files.
- The postconf command displays Postfix **main.cf** parameters: actual values, default values, or parameters that have non-default settings. This is a limited, preliminary utility. This program is likely to be superseded by something more powerful that can not only list but also edit the **main.cf** file.
- The postdrop command is the mail posting utility that is run by the sendmail command in order to deposit mail into the **maildrop** queue directory.
- The postkick command makes some internal communication channels available for use in, for example, shell scripts.
- The postlock command provides Postfix-compatible mailbox locking for use in, for example, shell scripts.
- The postlog command provides Postfix-compatible logging for shell scripts.
- The postmap command maintains Postfix lookup tables such as canonical, virtual and others. It is a cousin of the UNIX **makemap** command.
- The postqueue command is the utility that is run by the sendmail command in order to flush or list the mail queue.
- The postsuper command maintains the Postfix queue. It removes old temporary files, and moves queue files into the right directory after a change in the hashing depth of queue directories. This command is run at mail system startup time.

POSTFIX(1) POSTFIX(1)

NAME

postfix - Postfix control program

SYNOPSIS

postfix [**-Dv**] [**-c** *config_dir*] *command*

DESCRIPTION

This command is reserved for the superuser. To submit mail, use the Postfix **sendmail** command.

The **postfix** command controls the operation of the Postfix mail system: start or stop the **master** daemon, do a health check, and other maintenance.

The **postfix** command sets up a standardized environment and runs the **postfix-script** shell script to do the actual work.

The following commands are implemented:

check Validate the Postfix mail system configuration. Warn about bad directory/file ownership or permissions, and create missing directories.

start Start the Postfix mail system. This also runs the configuration check described above.

stop Stop the Postfix mail system in an orderly fashion. Running processes are allowed to terminate at their earliest convenience.

Note: in order to refresh the Postfix mail system after a configuration change, do not use the **start** and **stop** commands in succession. Use the **reload** command instead.

abort Stop the Postfix mail system abruptly. Running processes are signaled to stop immediately.

flush Force delivery: attempt to deliver every message in the deferred mail queue. Normally, attempts to deliver delayed mail happen at regular intervals, the interval doubling after each failed attempt.

reload Re-read configuration files. Running processes terminate at their earliest convenience.

The following options are implemented:

-c *config_dir*
Read the **main.cf** and **master.cf** configuration files in the named directory instead of the default configuration directory. Use this to distinguish between multiple Postfix instances on the same host.

-D (with **postfix start** only)
Run each Postfix daemon under control of a debugger as specified via the **debugger_command** configuration parameter.

-v Enable verbose logging for debugging purposes. Mul-

tiple **-v** options make the software increasingly verbose.

ENVIRONMENT

The **postfix** command exports the following environment variables before executing the **postfix-script** file:

MAIL_CONFIG
This is set when the -c command-line option is present.

MAIL_VERBOSE
This is set when the -v command-line option is present.

MAIL_DEBUG
This is set when the -D command-line option is present.

The following **main.cf** configuration parameters are exported as environment variables with the same names:

command_directory
Directory with Postfix administrative commands.

daemon_directory
Directory with Postfix daemon programs.

config_directory
Directory with Postfix configuration files and with administrative shell scripts.

queue_directory
The directory with Postfix queue files, with local inter-process communication endpoints, and with files needed for daemon programs that run in the optional chrooted environment.

mail_owner
The owner of Postfix queue files and of most Postfix processes.

setgid_group
The group for mail submission and queue management commands.

sendmail_path
The full pathname for the Postfix **sendmail** command.

newaliases_path
The full pathname for the Postfix **newaliases** command.

mailq_path
The full pathname for the Postfix **mailq** command.

manpage_directory
The directory for the Postfix on-line manual pages.

sample_directory
The directory for the Postfix sample configuration files.

readme_directory
The directory for the Postfix README files.

Other configuration parameters

import_environment
List of names of environment parameters that can be imported from non-Postfix processes.

FILES

$**config_directory/postfix-script**, administrative commands
$**config_directory/main.cf**, configuration parameters
$**config_directory/master.cf**, Postfix daemon processes

SEE ALSO

postconf(1) Postfix configuration management
postsuper(1) Postfix housekeeping
sendmail(1) Sendmail-compatible interface
postmap(1) Postfix lookup table management
master(8) Postfix master daemon
The respective manual pages for the daemon processes specified in the **master.cf** file, and the manual pages referenced by those manual pages.

LICENSE

The Secure Mailer license must be distributed with this software.

AUTHOR(S)

Wietse Venema
IBM T.J. Watson Research
P.O. Box 704
Yorktown Heights, NY 10598, USA

POSTFIX(1)

NAME

postalias - Postfix alias database maintenance

SYNOPSIS

postalias [**-Nfinorvw**] [**-c** *config_dir*] [**-d** *key*] [**-q** *key*] [*file_type*:]*file_name* ...

DESCRIPTION

The **postalias** command creates or queries one or more Postfix alias databases, or updates an existing one. The input and output file formats are expected to be compatible with Sendmail version 8, and are expected to be suitable for the use as NIS alias maps.

If the result files do not exist they will be created with the same group and other read permissions as the source file.

While a database update is in progress, signal delivery is postponed, and an exclusive, advisory, lock is placed on the entire database, in order to avoid surprises in spectator programs.

Options:

-N Include the terminating null character that terminates lookup keys and values. By default, Postfix does whatever is the default for the host operating system.

-c *config_dir*
Read the **main.cf** configuration file in the named directory instead of the default configuration directory.

-d *key* Search the specified maps for *key* and remove one entry per map. The exit status is zero when the requested information was found.

If a key value of **-** is specified, the program reads key values from the standard input stream. The exit status is zero when at least one of the requested keys was found.

-f Do not fold the lookup key to lower case while creating or querying a map.

-i Incremental mode. Read entries from standard input and do not truncate an existing database. By default, **postalias** creates a new database from the entries in *file_name*.

-n Don't include the terminating null character that terminates lookup keys and values. By default, Postfix does whatever is the default for the host operating system.

-o Do not release root privileges when processing a non-root input file. By default, **postalias** drops

root privileges and runs as the source file owner instead.

-q *key* Search the specified maps for *key* and print the first value found on the standard output stream. The exit status is zero when the requested information was found.

If a key value of **-** is specified, the program reads key values from the standard input stream and prints one line of *key: value* output for each key that was found. The exit status is zero when at least one of the requested keys was found.

-r When updating a table, do not warn about duplicate entries; silently replace them.

-v Enable verbose logging for debugging purposes. Multiple **-v** options make the software increasingly verbose.

-w When updating a table, do not warn about duplicate entries; silently ignore them.

Arguments:

file_type
The type of database to be produced.

btree The output is a btree file, named *file_name*.**db**. This is available only on systems with support for **db** databases.

dbm The output consists of two files, named *file_name*.**pag** and *file_name*.**dir**. This is available only on systems with support for **dbm** databases.

hash The output is a hashed file, named *file_name*.**db**. This is available only on systems with support for **db** databases.

Use the command **postconf -m** to find out what types of database your Postfix installation can support.

When no *file_type* is specified, the software uses the database type specified via the **default_database_type** configuration parameter. The default value for this parameter depends on the host environment.

file_name
The name of the alias database source file when creating a database.

DIAGNOSTICS

Problems are logged to the standard error stream. No output means no problems were detected. Duplicate entries are skipped and are flagged with a warning.

postalias terminates with zero exit status in case of success (including successful **postalias -q** lookup) and termi-

nates with non-zero exit status in case of failure.

ENVIRONMENT

MAIL_CONFIG
Directory with Postfix configuration files.

MAIL_VERBOSE
Enable verbose logging for debugging purposes.

CONFIGURATION PARAMETERS

The following **main.cf** parameters are especially relevant to this program. See the Postfix **main.cf** file for syntax details and for default values.

default_database_type
Default database type. On many UNIX systems, the default type is either **dbm** or **hash**.

berkeley_db_create_buffer_size
Amount of buffer memory to be used when creating a Berkeley DB **hash** or **btree** lookup table.

berkeley_db_read_buffer_size
Amount of buffer memory to be used when reading a Berkeley DB **hash** or **btree** lookup table.

STANDARDS

RFC 822 (ARPA Internet Text Messages)

SEE ALSO

aliases(5) format of alias database input file.
sendmail(1) mail posting and compatibility interface.

LICENSE

The Secure Mailer license must be distributed with this software.

AUTHOR(S)

Wietse Venema
IBM T.J. Watson Research
P.O. Box 704
Yorktown Heights, NY 10598, USA

POSTALIAS(1)

POSTCAT(1) POSTCAT(1)

NAME

postcat - show Postfix queue file contents

SYNOPSIS

postcat [**-v**] [*files*...]

DESCRIPTION

The **postcat** command prints the contents of the named Postfix queue *files* in human-readable form. If no *files* are specified on the command line, the program reads from standard input.

Options:

-v Enable verbose logging for debugging purposes. Multiple **-v** options make the software increasingly verbose.

DIAGNOSTICS

Problems are reported to the standard error stream.

LICENSE

The Secure Mailer license must be distributed with this software.

AUTHOR(S)

Wietse Venema
IBM T.J. Watson Research
P.O. Box 704
Yorktown Heights, NY 10598, USA

POSTCAT(1)

SENDMAIL(1) SENDMAIL(1)

NAME

sendmail - Postfix to Sendmail compatibility interface

SYNOPSIS

sendmail [*option ...*] [*recipient ...*]

mailq
sendmail -bp

newaliases
sendmail -I

DESCRIPTION

The **sendmail** program implements the Postfix to Sendmail compatibility interface. For the sake of compatibility with existing applications, some Sendmail command-line options are recognized but silently ignored.

By default, **sendmail** reads a message from standard input until EOF or until it reads a line with only a . character, and arranges for delivery. **sendmail** relies on the **postdrop**(1) command to create a queue file in the **maildrop** directory.

Specific command aliases are provided for other common modes of operation:

mailq List the mail queue. Each entry shows the queue file ID, message size, arrival time, sender, and the recipients that still need to be delivered. If mail could not be delivered upon the last attempt, the reason for failure is shown. This mode of operation is implemented by executing the **postqueue**(1) command.

newaliases
Initialize the alias database. If no input file is specified (with the **-oA** option, see below), the program processes the file(s) specified with the **alias_database** configuration parameter. If no alias database type is specified, the program uses the type specified with the **default_database_type** configuration parameter. This mode of operation is implemented by running the **postalias**(1) command.

Note: it may take a minute or so before an alias database update becomes visible. Use the **postfix reload** command to eliminate this delay.

These and other features can be selected by specifying the appropriate combination of command-line options. Some features are controlled by parameters in the **main.cf** configuration file.

The following options are recognized:

-Am (ignored)

-Ac (ignored)
Postfix sendmail uses the same configuration file regardless of whether or not a message is an ini-

tial submission.

-B *body_type*
The message body MIME type: **7BIT** or **8BITMIME**.

-C *config_file* (ignored :-)
The path name of the **sendmail.cf** file. Postfix configuration files are kept in **/etc/postfix**.

-F *full_name*
Set the sender full name. This is used only with messages that have no **From:** message header.

-G (ignored)
Gateway (relay) submission, as opposed to initial user submission.

-I Initialize alias database. See the **newaliases** command above.

-L *label* (ignored)
The logging label. Use the **syslog_name** configuration parameter instead.

-N *dsn* (ignored)
Delivery status notification control. Currently, Postfix does not implement **DSN**.

-R *return_limit* (ignored)
Limit the size of bounced mail. Use the **bounce_size_limit** configuration parameter instead.

-X *log_file* (ignored)
Log mailer traffic. Use the **debug_peer_list** and **debug_peer_level** configuration parameters instead.

-U (ignored)
Initial user submission.

-V Variable Envelope Return Path. Given an envelope sender address of the form *owner-listname*@*origin*, each recipient *user*@*domain* receives mail with a personalized envelope sender address.

By default, the personalized envelope sender address is *owner-listname*+*user*=*domain*@*origin*. The default **+** and **=** characters are configurable with the **default_verp_delimiters** configuration parameter.

-V*xy* As **-V**, but uses *x* and *y* as the VERP delimiter characters, instead of the characters specified with the **default_verp_delimiters** configuration parameter.

-bd Go into daemon mode. This mode of operation is implemented by executing the **postfix start** command.

-bi Initialize alias database. See the **newaliases** command above.

-bm Read mail from standard input and arrange for

delivery. This is the default mode of operation.

-bp List the mail queue. See the **mailq** command above.

-bs Stand-alone SMTP server mode. Read SMTP commands from standard input, and write responses to standard output. In stand-alone SMTP server mode, UCE restrictions and access controls are disabled by default. To enable them, run the process as the **mail_owner** user.

This mode of operation is implemented by running the **smtpd**(8) daemon.

-f *sender*
Set the envelope sender address. This is the address where delivery problems are sent to, unless the message contains an **Errors-To:** message header.

-h *hop_count* (ignored)
Hop count limit. Use the **hopcount_limit** configuration parameter instead.

-i When reading a message from standard input, don't treat a line with only a . character as the end of input.

-m (ignored)
Backwards compatibility.

-n (ignored)
Backwards compatibility.

-oA*alias_database*
Non-default alias database. Specify *pathname* or *type*:*pathname*. See **postalias**(1) for details.

-o7 (ignored)

-o8 (ignored)
To send 8-bit or binary content, use an appropriate MIME encapsulation and specify the appropriate **-B** command-line option.

-oi When reading a message from standard input, don't treat a line with only a . character as the end of input.

-om (ignored)
The sender is never eliminated from alias etc. expansions.

-o *x value* (ignored)
Set option *x* to *value*. Use the equivalent configuration parameter in **main.cf** instead.

-r *sender*
Set the envelope sender address. This is the address where delivery problems are sent to, unless the message contains an **Errors-To:** message header.

-q Attempt to deliver all queued mail. This is imple-

mented by executing the **postqueue**(1) command.

-q*interval* (ignored)
The interval between queue runs. Use the **queue_run_delay** configuration parameter instead.

-qR*site*
Schedule immediate delivery of all mail that is queued for the named *site*. This option accepts only *site* names that are eligible for the "fast flush" service, and is implemented by executing the **postqueue**(1) command. See **flush**(8) for more information about the "fast flush" service.

-qS*site*
This command is not implemented. Use the slower **sendmail -q** command instead.

-t Extract recipients from message headers. This requires that no recipients be specified on the command line.

-v Enable verbose logging for debugging purposes. Multiple **-v** options make the software increasingly verbose. For compatibility with mailx and other mail submission software, a single **-v** option produces no output.

SECURITY
By design, this program is not set-user (or group) id. However, it must handle data from untrusted users or untrusted machines. Thus, the usual precautions need to be taken against malicious inputs.

DIAGNOSTICS
Problems are logged to **syslogd**(8) and to the standard error stream.

ENVIRONMENT
MAIL_CONFIG
Directory with Postfix configuration files.

MAIL_VERBOSE
Enable verbose logging for debugging purposes.

MAIL_DEBUG
Enable debugging with an external command, as specified with the **debugger_command** configuration parameter.

FILES
/var/spool/postfix, mail queue
/etc/postfix, configuration files

CONFIGURATION PARAMETERS
See the Postfix **main.cf** file for syntax details and for default values. Use the **postfix reload** command after a configuration change.

alias_database
Default alias database(s) for **newaliases**. The default value for this parameter is system-spe-

cific.

bounce_size_limit
The amount of original message context that is sent along with a non-delivery notification.

default_database_type
Default alias etc. database type. On many UNIX systems the default type is either **dbm** or **hash**.

debugger_command
Command that is executed after a Postfix daemon has initialized.

debug_peer_level
Increment in verbose logging level when a remote host matches a pattern in the **debug_peer_list** parameter.

debug_peer_list
List of domain or network patterns. When a remote host matches a pattern, increase the verbose logging level by the amount specified in the **debug_peer_level** parameter.

default_verp_delimiters
The VERP delimiter characters that are used when the **-V** command line option is specified without delimiter characters.

fast_flush_domains
List of domains that will receive "fast flush" service (default: all domains that this system is willing to relay mail to). This list specifies the domains that Postfix accepts in the SMTP **ETRN** request and in the **sendmail -qR** command.

fork_attempts
Number of attempts to **fork**() a process before giving up.

fork_delay
Delay in seconds between successive **fork**()attempts.

hopcount_limit
Limit the number of **Received:** message headers.

mail_owner
The owner of the mail queue and of most Postfix processes.

command_directory
Directory with Postfix support commands.

daemon_directory
Directory with Postfix daemon programs.

queue_directory
Top-level directory of the Postfix queue. This is also the root directory of Postfix daemons that run chrooted.

queue_run_delay
The time between successive scans of the deferred queue.

verp_delimiter_filter
The characters that Postfix accepts as VERP delimiter characters.

POSTCONF(1) POSTCONF(1)

NAME

postconf - Postfix configuration utility

SYNOPSIS

postconf [**-dhmlnv**] [**-c** *config_dir*] [*parameter ...*]

postconf [**-ev**] [**-c** *config_dir*] [*parameter=value ...*]

DESCRIPTION

The **postconf** command prints the actual value of *parameter* (all known parameters by default) one parameter per line, changes its value, or prints other information about the Postfix mail system.

Options:

-c *config_dir*
The **main.cf** configuration file is in the named directory instead of the default configuration directory.

-d Print default parameter settings instead of actual settings.

-e Edit the **main.cf** configuration file. The file is copied to a temporary file then renamed into place. Parameters and values are specified on the command line. Use quotes in order to protect shell metacharacters and whitespace.

-h Show parameter values only, not the ``name = '' label that normally precedes the value.

-l List the names of all supported mailbox locking methods. Postfix supports the following methods:

flock A kernel-based advisory locking method for local files only. This locking method is available only on systems with a BSD compatible library.

fcntl A kernel-based advisory locking method for local and remote files.

dotlock An application-level locking method. An application locks a file named *filename* by creating a file named *filename*.**lock**. The application is expected to remove its own lock file, as well as stale lock files that were left behind after abnormal termination.

-m List the names of all supported lookup table types. Postfix lookup tables are specified as *type*:*name*, where *type* is one of the types listed below. The table *name* syntax depends on the lookup table type.

btree A sorted, balanced tree structure. This is available only on systems with support for Berkeley DB databases.

dbm An indexed file type based on hashing. This is available only on systems with support for DBM databases.

environ
The UNIX process environment array. The lookup key is the variable name. Originally

implemented for testing, someone may find this useful someday.

hash An indexed file type based on hashing. This is available only on systems with support for Berkeley DB databases.

ldap (read-only)
Perform lookups using the LDAP protocol. This is described in an LDAP_README file.

mysql (read-only)
Perform lookups using the MYSQL protocol. This is described in a MYSQL_README file.

pcre (read-only)
A lookup table based on Perl Compatible Regular Expressions. The file format is described in **pcre_table**(5).

proxy (read-only)
A lookup table that is implemented via the Postfix **proxymap**(8) service. The table name syntax is *type*:*name*.

regexp (read-only)
A lookup table based on regular expressions. The file format is described in **regexp_table**(5).

static (read-only)
A table that always returns its name as lookup result. For example, **static:foobar** always returns the string **foobar** as lookup result.

unix (read-only)
A limited way to query the UNIX authentication database. The following tables are implemented:

unix:passwd.byname
The table is the UNIX password database. The key is a login name. The result is a password file entry in passwd(5) format.

unix:group.byname
The table is the UNIX group database. The key is a group name. The result is a group file entry in group(5) format.

Other table types may exist depending on how Postfix was built.

-n Print non-default parameter settings only.

-v Enable verbose logging for debugging purposes. Multiple **-v** options make the software increasingly verbose.

DIAGNOSTICS Problems are reported to the standard error stream.

ENVIRONMENT

MAIL_CONFIG Directory with Postfix configuration files.

POSTDROP(1) POSTDROP(1)

NAME

postdrop - Postfix mail posting utility

SYNOPSIS

postdrop [**-rv**] [**-c** *config_dir*]

DESCRIPTION

The **postdrop** command creates a file in the **maildrop** directory and copies its standard input to the file.

Options:

-c The **main.cf** configuration file is in the named directory instead of the default configuration directory. See also the MAIL_CONFIG environment setting below.

-r Use a Postfix-internal protocol for reading the message from standard input, and for reporting status information on standard output. This is currently the only supported method.

-v Enable verbose logging for debugging purposes. Multiple **-v** options make the software increasingly verbose.

SECURITY

The command is designed to run with set-group ID privileges, so that it can write to the **maildrop** queue directory and so that it can connect to Postfix daemon processes.

DIAGNOSTICS

Fatal errors: malformed input, I/O error, out of memory. Problems are logged to **syslogd**(8) and to the standard error stream. When the input is incomplete, or when the process receives a HUP, INT, QUIT or TERM signal, the queue file is deleted.

ENVIRONMENT

MAIL_CONFIG

Directory with the **main.cf** file. In order to avoid exploitation of set-group ID privileges, it is not possible to specify arbitrary directory names.

A non-standard directory is allowed only if the name is listed in the standard **main.cf** file, in the **alternate_config_directories** configuration parameter value.

Only the superuser is allowed to specify arbitrary directory names.

FILES

/var/spool/postfix, mail queue
/etc/postfix, configuration files

CONFIGURATION PARAMETERS

See the Postfix **main.cf** file for syntax details and for default values. Use the **postfix reload** command after a configuration change.

import_environment
List of names of environment parameters that can be imported from non-Postfix processes.

queue_directory
Top-level directory of the Postfix queue. This is also the root directory of Postfix daemons that run chrooted.

SEE ALSO
sendmail(1) compatibility interface
syslogd(8) system logging

LICENSE
The Secure Mailer license must be distributed with this software.

AUTHOR(S)
Wietse Venema
IBM T.J. Watson Research
P.O. Box 704
Yorktown Heights, NY 10598, USA

POSTDROP(1)

POSTKICK(1) POSTKICK(1)

NAME

postkick - kick a Postfix service

SYNOPSIS

postkick [**-c** *config_dir*] [**-v**] *class service request*

DESCRIPTION

The **postkick** command sends *request* to the specified *service* over a local transport channel. This command makes Postfix private IPC accessible for use in, for example, shell scripts.

Options:

-c *config_dir*
Read the **main.cf** configuration file in the named directory instead of the default configuration directory.

-v Enable verbose logging for debugging purposes. Multiple **-v** options make the software increasingly verbose.

Arguments:

class Name of a class of local transport channel endpoints, either **public** (accessible by any local user) or **private** (administrative access only).

service
The name of a local transport endpoint within the named class.

request
A string. The list of valid requests is service-specific.

DIAGNOSTICS

Problems and transactions are logged to the standard error stream.

ENVIRONMENT

MAIL_CONFIG
Directory with Postfix configuration files.

MAIL_VERBOSE
Enable verbose logging for debugging purposes.

CONFIGURATION PARAMETERS

The following **main.cf** parameters are especially relevant to this program. See the Postfix **main.cf** file for syntax details and for default values.

queue_directory
Location of the Postfix queue, and of the local IPC communication endpoints.

POSTLOCK(1) POSTLOCK(1)

NAME

postlock - lock mail folder and execute command

SYNOPSIS

postlock [**-c** *config_dir*] [**-l** *lock_style*] [**-v**] *file command...*

DESCRIPTION

The **postlock** command locks *file* for exclusive access, and executes *command*. The locking method is compatible with the Postfix UNIX-style local delivery agent.

Options:

-c *config_dir*
Read the **main.cf** configuration file in the named directory instead of the default configuration directory.

-l *lock_style*
Override the locking method specified via the **mailbox_delivery_lock** configuration parameter (see below).

-v Enable verbose logging for debugging purposes. Multiple **-v** options make the software increasingly verbose.

Arguments:

file A mailbox file. The user should have read/write permission.

command...
The command to execute while *file* is locked for exclusive access. The command is executed directly, i.e. without interpretation by a shell command interpreter.

DIAGNOSTICS

The result status is 75 (EX_TEMPFAIL) when **postlock** could not perform the requested operation. Otherwise, the exit status is the exit status from the command.

BUGS

With remote file systems, the ability to acquire a lock does not necessarily eliminate access conflicts. Avoid file access by processes running on different machines.

ENVIRONMENT

MAIL_CONFIG
Directory with Postfix configuration files.

MAIL_VERBOSE
Enable verbose logging for debugging purposes.

CONFIGURATION PARAMETERS

The following **main.cf** parameters are especially relevant to this program. See the Postfix **main.cf** file for syntax details and for default values.

Locking controls

deliver_lock_attempts
Limit the number of attempts to acquire an exclusive lock.

deliver_lock_delay
Time in seconds between successive attempts to acquire an exclusive lock.

stale_lock_time
Limit the time after which a stale lock is removed.

mailbox_delivery_lock
What file locking method(s) to use when delivering to a UNIX-style mailbox. The default setting is system dependent. For a list of available file locking methods, use the **postconf -l** command.

Resource controls

fork_attempts
Number of attempts to **fork**() a process before giving up.

fork_delay
Delay in seconds between successive **fork**() attempts.

LICENSE

The Secure Mailer license must be distributed with this software.

AUTHOR(S)

Wietse Venema
IBM T.J. Watson Research
P.O. Box 704
Yorktown Heights, NY 10598, USA

POSTLOG(1) POSTLOG(1)

NAME

postlog - Postfix-compatible logging utility

SYNOPSIS

postlog [**-iv**] [**-c** *config_dir*] [**-p** *priority*] [**-t** *tag*] [*text...*]

DESCRIPTION

The **postlog** command implements a Postfix-compatible logging interface for use in, for example, shell scripts.

By default, **postlog** logs the *text* given on the command line as one record. If no *text* is specified on the command line, **postlog** reads from standard input and logs each input line as one record.

Logging is sent to **syslogd**(8); when the standard error stream is connected to a terminal, logging is sent there as well.

The following options are implemented:

-c *config_dir*
Read the **main.cf** configuration file in the named directory instead of the default configuration directory.

-i Include the process ID in the logging tag.

-p *priority*
Specifies the logging severity: **info** (default), **warn**, **error**, **fatal**, or **panic**.

-t *tag* Specifies the logging tag, that is, the identifying name that appears at the beginning of each logging record.

-v Enable verbose logging for debugging purposes. Multiple **-v** options make the software increasingly verbose.

SEE ALSO

syslogd(8) syslog daemon.

LICENSE

The Secure Mailer license must be distributed with this software.

AUTHOR(S)

Wietse Venema
IBM T.J. Watson Research
P.O. Box 704
Yorktown Heights, NY 10598, USA

1

NAME

postmap - Postfix lookup table management

SYNOPSIS

postmap [**-Nfinorvw**] [**-c** *config_dir*] [**-d** *key*] [**-q** *key*] [*file_type*:]*file_name* ...

DESCRIPTION

The **postmap** command creates or queries one or more Postfix lookup tables, or updates an existing one. The input and output file formats are expected to be compatible with:

makemap *file_type file_name* < *file_name*

If the result files do not exist they will be created with the same group and other read permissions as the source file.

While the table update is in progress, signal delivery is postponed, and an exclusive, advisory, lock is placed on the entire table, in order to avoid surprises in spectator programs.

The format of a lookup table input file is as follows:

- A table entry has the form

  '*key* whitespace *value*'

- Empty lines and whitespace-only lines are ignored, as are lines whose first non-whitespace character is a `#'.

- A logical line starts with non-whitespace text. A line that starts with whitespace continues a logical line.

The *key* and *value* are processed as is, except that surrounding white space is stripped off. Unlike with Postfix alias databases, quotes cannot be used to protect lookup keys that contain special characters such as `#' or whitespace. The *key* is mapped to lowercase to make mapping lookups case insensitive.

Options:

-N Include the terminating null character that terminates lookup keys and values. By default, Postfix does whatever is the default for the host operating system.

-c *config_dir*
Read the **main.cf** configuration file in the named directory instead of the default configuration directory.

-d *key* Search the specified maps for *key* and remove one entry per map. The exit status is zero when the requested information was found.

If a key value of **-** is specified, the program reads

key values from the standard input stream. The exit status is zero when at least one of the requested keys was found.

-f Do not fold the lookup key to lower case while creating or querying a map.

-i Incremental mode. Read entries from standard input and do not truncate an existing database. By default, **postmap** creates a new database from the entries in **file_name**.

-n Don't include the terminating null character that terminates lookup keys and values. By default, Postfix does whatever is the default for the host operating system.

-o Do not release root privileges when processing a non-root input file. By default, **postmap** drops root privileges and runs as the source file owner instead.

-q *key* Search the specified maps for *key* and print the first value found on the standard output stream. The exit status is zero when the requested information was found.

If a key value of **-** is specified, the program reads key values from the standard input stream and prints one line of *key value* output for each key that was found. The exit status is zero when at least one of the requested keys was found.

-r When updating a table, do not warn about duplicate entries; silently replace them.

-v Enable verbose logging for debugging purposes. Multiple **-v** options make the software increasingly verbose.

-w When updating a table, do not warn about duplicate entries; silently ignore them.

Arguments:

file_type
The type of database to be produced.

btree The output file is a btree file, named *file_name*.**db**. This is available only on systems with support for **db** databases.

dbm The output consists of two files, named *file_name*.**pag** and *file_name*.**dir**. This is available only on systems with support for **dbm** databases.

hash The output file is a hashed file, named *file_name*.**db**. This is available only on systems with support for **db** databases.

Use the command **postconf -m** to find out what types of database your Postfix installation can support.

When no *file_type* is specified, the software uses the database type specified via the **default_database_type** configuration parameter.

file_name
The name of the lookup table source file when rebuilding a database.

DIAGNOSTICS

Problems and transactions are logged to the standard error stream. No output means no problems. Duplicate entries are skipped and are flagged with a warning.

postmap terminates with zero exit status in case of success (including successful **postmap -q** lookup) and terminates with non-zero exit status in case of failure.

ENVIRONMENT

MAIL_CONFIG
Directory with Postfix configuration files.

MAIL_VERBOSE
Enable verbose logging for debugging purposes.

CONFIGURATION PARAMETERS

default_database_type
Default output database type. On many UNIX systems, the default database type is either **hash** or **dbm**.

berkeley_db_create_buffer_size
Amount of buffer memory to be used when creating a Berkeley DB **hash** or **btree** lookup table.

berkeley_db_read_buffer_size
Amount of buffer memory to be used when reading a Berkeley DB **hash** or **btree** lookup table.

LICENSE

The Secure Mailer license must be distributed with this software.

AUTHOR(S)

Wietse Venema
IBM T.J. Watson Research
P.O. Box 704
Yorktown Heights, NY 10598, USA

POSTMAP(1)

NAME

postqueue - Postfix queue control

SYNOPSIS

postqueue [**-c** *config_dir*] **-f**
postqueue [**-c** *config_dir*] **-p**
postqueue [**-c** *config_dir*] **-s** *site*

DESCRIPTION

The **postqueue** program implements the Postfix user interface for queue management. It implements operations that are traditionally available via the **sendmail**(1) command. See the **postsuper**(1) command for queue operations that require super-user privileges such as deleting a message from the queue or changing the status of a message.

The following options are recognized:

-c *config_dir*
The **main.cf** configuration file is in the named directory instead of the default configuration directory. See also the MAIL_CONFIG environment setting below.

-f Flush the queue: attempt to deliver all queued mail.

This option implements the traditional **sendmail -q** command, by contacting the Postfix **qmgr**(8) daemon.

-p Produce a traditional sendmail-style queue listing. This option implements the traditional **mailq** command, by contacting the Postfix **showq**(8) daemon.

Each queue entry shows the queue file ID, message size, arrival time, sender, and the recipients that still need to be delivered. If mail could not be delivered upon the last attempt, the reason for failure is shown. This mode of operation is implemented by executing the **postqueue**(1) command. The queue ID string is followed by an optional status character:

***** The message is in the **active** queue, i.e. the message is selected for delivery.

! The message is in the **hold** queue, i.e. no further delivery attempt will be made until the mail is taken off hold.

-s *site*
Schedule immediate delivery of all mail that is queued for the named *site*. The site must be eligible for the "fast flush" service. See **flush**(8) for more information about the "fast flush" service.

This option implements the traditional **sendmail -qR***site* command, by contacting the Postfix **flush**(8) daemon.

-v Enable verbose logging for debugging purposes. Multiple **-v** options make the software increasingly

verbose.

SECURITY

This program is designed to run with set-group ID privileges, so that it can connect to Postfix daemon processes.

DIAGNOSTICS

Problems are logged to **syslogd**(8) and to the standard error stream.

ENVIRONMENT

MAIL_CONFIG

Directory with the **main.cf** file.

In order to avoid exploitation of set-group ID privileges, it is not possible to specify arbitrary directory names.

A non-standard directory is allowed only if the name is listed in the standard **main.cf** file, in the **alternate_config_directories** configuration parameter value.

Only the superuser is allowed to specify arbitrary directory names.

FILES

/var/spool/postfix, mail queue
/etc/postfix, configuration files

CONFIGURATION PARAMETERS

import_environment

List of names of environment parameters that can be imported from non-Postfix processes.

queue_directory

Top-level directory of the Postfix queue. This is also the root directory of Postfix daemons that run chrooted.

fast_flush_domains

List of domains that will receive "fast flush" service (default: all domains that this system is willing to relay mail to). This list specifies the domains that Postfix accepts in the SMTP **ETRN** request and in the **sendmail -qR** command.

SEE ALSO

sendmail(1) sendmail-compatible user interface
postsuper(1) privileged queue operations
qmgr(8) queue manager
showq(8) list mail queue
flush(8) fast flush service

LICENSE

The Secure Mailer license must be distributed with this software.

AUTHOR(S)

Wietse Venema
IBM T.J. Watson Research
P.O. Box 704
Yorktown Heights, NY 10598, USA

POSTQUEUE(1)

POSTSUPER(1) POSTSUPER(1)

NAME

postsuper - Postfix superintendent

SYNOPSIS

postsuper [**-psv**] [**-c** *config_dir*] [**-d** *queue_id*] [**-h** *queue_id*] [**-H** *queue_id*] [**-r** *queue_id*] [*directory ...*]

DESCRIPTION

The **postsuper** command does maintenance jobs on the Postfix queue. Use of the command is restricted to the superuser. See the **postqueue** command for unprivileged queue operations such as listing or flushing the mail queue.

By default, **postsuper** performs the operations requested with the **-s** and **-p** command-line options on all Postfix queue directories - this includes the **incoming**, **active** and **deferred** directories with mail files and the **bounce**, **defer** and **flush** directories with log files.

Options:

-c *config_dir*

The **main.cf** configuration file is in the named directory instead of the default configuration directory. See also the MAIL_CONFIG environment setting below.

-d *queue_id*

Delete one message with the named queue ID from the named mail queue(s) (default: **hold**, **incoming**, **active** and **deferred**). If a *queue_id* of **-** is specified, the program reads queue IDs from standard input. For example, to delete all mail from or to **user@example.com**:

```
mailq | tail +2 | awk  'BEGIN { RS = "" } \
    / user@example\.com$/ { print $1 } \
' | tr -d '*!' | postsuper -d -
```

Specify **-d ALL** to remove all messages; for example, specify **-d ALL deferred** to delete mail in the **deferred** queue. As a safety measure, the word **ALL** must be specified in upper case.

Postfix queue IDs are reused. There is a very small possibility that postsuper deletes the wrong message file when it is executed while the Postfix mail system is running.

The scenario is as follows:

1) The Postfix queue manager deletes the message that **postsuper** is supposed to delete, because Postfix is finished with the message.

2) New mail arrives, and the new message is given the same queue ID as the message that **postsuper** is supposed to delete. The proba-

bility for reusing a deleted queue ID is about 1 in 2**15 (the number of different microsecond values that the system clock can distinguish within a second).

3) **postsuper** deletes the new message, instead of the old message that it should have deleted.

-h *queue_id*
Put mail "on hold" so that no attempt is made to deliver it. Move one message with the named queue ID from the named mail queue(s) (default: **incoming**, **active** and **deferred**) to the **hold** queue. If a *queue_id* of **-** is specified, the program reads queue IDs from standard input.

Specify **-h ALL** to hold all messages; for example, specify **-h ALL deferred** to hold mail in the **deferred** queue. As a safety measure, the word **ALL** must be specified in upper case.

Note: mail that is put "on hold" will not expire when its time in the queue exceeds the **maximal_queue_lifetime** setting.

-H *queue_id*
Release mail that was put "on hold". Move one message with the named queue ID from the named mail queue(s) (default: **hold**) to the **deferred** queue. If a *queue_id* of **-** is specified, the program reads queue IDs from standard input.

Specify **-H ALL** to release all mail that is "on hold". As a safety measure, the word **ALL** must be specified in upper case.

-p Purge old temporary files that are left over after system or software crashes.

-r *queue_id*
Requeue the message with the named queue ID from the named mail queue(s) (default: **hold**, **incoming**, **active** and **deferred**). To requeue multiple messages, specify multiple **-r** command-line options. Alternatively, if a *queue_id* of **-** is specified, the program reads queue IDs from standard input.

Specify **-r ALL** to requeue all messages. As a safety measure, the word **ALL** must be specified in upper case.

A requeued message is moved to the **maildrop** queue, from where it is copied by the pickup daemon to a new file whose name is guaranteed to match the new queue file inode number. The new queue file is subjected again to mail address rewriting and substitution. This is useful when rewriting rules or virtual mappings have changed.

Postfix queue IDs are reused. There is a very small possibility that **postsuper** requeues the wrong message file when it is executed while the Postfix

```
mail system is running, but no harm should be done.
```

-s Structure check and structure repair. It is highly recommended to perform this operation once before Postfix startup.

- Rename files whose name does not match the message file inode number. This operation is necessary after restoring a mail queue from a different machine, or from backup media.
- Move queue files that are in the wrong place in the file system hierarchy and remove sub-directories that are no longer needed. File position rearrangements are necessary after a change in the **hash_queue_names** and/or **hash_queue_depth** configuration parameters.

-v Enable verbose logging for debugging purposes. Multiple **-v** options make the software increasingly verbose.

DIAGNOSTICS

Problems are reported to the standard error stream and to **syslogd**.

postsuper reports the number of messages deleted with **-d**, the number of messages requeued with **-r**, and the number of messages whose queue file name was fixed with **-s**. The report is written to the standard error stream and to **syslogd**.

ENVIRONMENT

MAIL_CONFIG
Directory with the **main.cf** file.

BUGS

Mail that is not sanitized by Postfix (i.e. mail in the **maildrop** queue) cannot be placed "on hold".

CONFIGURATION PARAMETERS

See the Postfix **main.cf** file for syntax details and for default values.

hash_queue_depth
Number of subdirectory levels for hashed queues.

hash_queue_names
The names of queues that are organized into multiple levels of subdirectories.

SEE ALSO

sendmail(1) sendmail-compatible user interface
postqueue(1) unprivileged queue operations

LICENSE

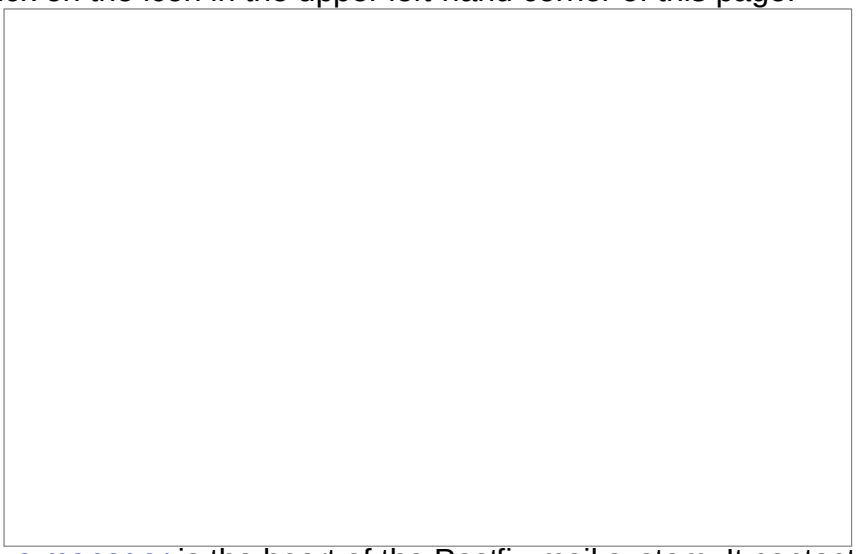
The Secure Mailer license must be distributed with this software.

AUTHOR(S)

Wietse Venema
IBM T.J. Watson Research
P.O. Box 704
Yorktown Heights, NY 10598, USA

Delivering Mail

Once a message has reached the **incoming** queue the next step is to deliver it. The figure shows the main components of the Postfix mail delivery apparatus. For an explanation of the symbols, click on the icon in the upper left-hand corner of this page.



The queue manager is the heart of the Postfix mail system. It contacts the local, smtp, lmtp, or pipe delivery agents, and sends a delivery request with queue file pathname information, the message sender address, the host to deliver to if the destination is remote, and one or more message recipient addresses.

The queue manager maintains a separate **deferred** queue for mail that cannot be delivered, so that a large mail backlog will not slow down normal queue accesses.

The queue manager maintains a small **active** queue with just the few messages that it has opened for delivery. The **active** queue acts as a limited window on the potentially much larger **incoming** or **deferred** queues. The small **active** queue prevents the queue manager from running out of memory under heavy load.

Optionally, the queue manager bounces mail for recipients that are listed in the relocated table. This table contains contact information for users or even entire domains that no longer exist.

- On request by the queue manager, the trivial-rewrite daemon resolves destinations. By default, it only distinguishes between *local* and *remote* destinations. Additional routing information can be specified with the optional transport table.
- On request by the queue manager, the bounce or defer daemon generates non-delivery reports when mail cannot be delivered, either due to an unrecoverable error or because the destination is unreachable for an extended period of time.
- The local delivery agent understands UNIX-style mailboxes, **sendmail**-style system-wide alias databases, and **sendmail**-style per-user .forward files. Multiple local delivery agents can be run in parallel, but parallel delivery to the same user is usually limited.

  Together with the sendmail mail posting agent, the local delivery agent implements the familiar Sendmail user interface.

- The local delivery agent has hooks for alternative forms of local delivery: you can configure it to deliver to mailbox files in user home directories, and you can even configure it to delegate mailbox delivery to an external command such as the popular procmail program.
- The virtual delivery agent is a very much stripped down version of the local delivery agent that delivers to mailboxes only. This is the most secure Postfix delivery agent, because it does not aliases expansions and no .forward file expansions.

  This delivery agent can deliver mail for multiple domains, which makes it especially suitable for hosting lots of small domains on a single machine.
- The SMTP client looks up a list of mail exchangers for the destination host, sorts the list by preference, and tries each address in turn until it finds a server that responds. On a busy Postfix system you will see several SMTP client processes running in parallel.
- The LMTP client speaks a protocol similar to SMTP. The client can connect to local or remote mailbox servers such as Cyrus. All the queue management is done by Postfix. The advantage of this setup is that one Postfix machine can feed multiple mailbox servers over LMTP. The opposite is true as well: one mailbox server can be fed over LMTP by multiple Postfix machines.
- The pipe mailer is the outbound interface to other mail transports (the sendmail program is the inbound interface). The Postfix mail system comes with examples for delivery via the **UUCP** protocol. At the time of writing, this venerable protocol is still widely used. By default, Postfix understands bang path style addresses.

QMGR(8) QMGR(8)

NAME

qmgr - Postfix queue manager

SYNOPSIS

qmgr [generic Postfix daemon options]

DESCRIPTION

The **qmgr** daemon awaits the arrival of incoming mail and arranges for its delivery via Postfix delivery processes. The actual mail routing strategy is delegated to the **trivial-rewrite**(8) daemon. This program expects to be run from the **master**(8) process manager.

Mail addressed to the local **double-bounce** address is silently discarded. This stops potential loops caused by undeliverable bounce notifications.

MAIL QUEUES

The **qmgr** daemon maintains the following queues:

incoming
Inbound mail from the network, or mail picked up by the local **pickup** agent from the **maildrop** directory.

active Messages that the queue manager has opened for delivery. Only a limited number of messages is allowed to enter the **active** queue (leaky bucket strategy, for a fixed delivery rate).

deferred
Mail that could not be delivered upon the first attempt. The queue manager implements exponential backoff by doubling the time between delivery attempts.

corrupt
Unreadable or damaged queue files are moved here for inspection.

hold Messages that are kept "on hold" are kept here until someone sets them free.

DELIVERY STATUS REPORTS

The **qmgr** daemon keeps an eye on per-message delivery status reports in the following directories. Each status report file has the same name as the corresponding message file:

bounce Per-recipient status information about why mail is bounced. These files are maintained by the **bounce**(8) daemon.

defer Per-recipient status information about why mail is delayed. These files are maintained by the **defer**(8) daemon.

The **qmgr** daemon is responsible for asking the **bounce**(8) or **defer**(8) daemons to send non-delivery reports.

STRATEGIES

The queue manager implements a variety of strategies for either opening queue files (input) or for message delivery (output).

leaky bucket

This strategy limits the number of messages in the **active** queue and prevents the queue manager from running out of memory under heavy load.

fairness

When the **active** queue has room, the queue manager takes one message from the **incoming** queue and one from the **deferred** queue. This prevents a large mail backlog from blocking the delivery of new mail.

slow start

This strategy eliminates "thundering herd" problems by slowly adjusting the number of parallel deliveries to the same destination.

round robin

The queue manager sorts delivery requests by destination. Round-robin selection prevents one destination from dominating deliveries to other destinations.

exponential backoff

Mail that cannot be delivered upon the first attempt is deferred. The time interval between delivery attempts is doubled after each attempt.

destination status cache

The queue manager avoids unnecessary delivery attempts by maintaining a short-term, in-memory list of unreachable destinations.

TRIGGERS

On an idle system, the queue manager waits for the arrival of trigger events, or it waits for a timer to go off. A trigger is a one-byte message. Depending on the message received, the queue manager performs one of the following actions (the message is followed by the symbolic constant used internally by the software):

D (QMGR_REQ_SCAN_DEFERRED)

Start a deferred queue scan. If a deferred queue scan is already in progress, that scan will be restarted as soon as it finishes.

I (QMGR_REQ_SCAN_INCOMING)

Start an incoming queue scan. If an incoming queue scan is already in progress, that scan will be restarted as soon as it finishes.

A (QMGR_REQ_SCAN_ALL)

Ignore deferred queue file time stamps. The request affects the next deferred queue scan.

F (QMGR_REQ_FLUSH_DEAD)

Purge all information about dead transports and destinations.

W (TRIGGER_REQ_WAKEUP)
Wakeup call, This is used by the master server to instantiate servers that should not go away forever. The action is to start an incoming queue scan.

The **qmgr** daemon reads an entire buffer worth of triggers. Multiple identical trigger requests are collapsed into one, and trigger requests are sorted so that **A** and **F** precede **D** and **I**. Thus, in order to force a deferred queue run, one would request **A F D**; in order to notify the queue manager of the arrival of new mail one would request **I**.

STANDARDS
None. The **qmgr** daemon does not interact with the outside world.

SECURITY
The **qmgr** daemon is not security sensitive. It reads single-character messages from untrusted local users, and thus may be susceptible to denial of service attacks. The **qmgr** daemon does not talk to the outside world, and it can be run at fixed low privilege in a chrooted environment.

DIAGNOSTICS
Problems and transactions are logged to the syslog daemon. Corrupted message files are saved to the **corrupt** queue for further inspection.

Depending on the setting of the **notify_classes** parameter, the postmaster is notified of bounces and of other trouble.

BUGS
A single queue manager process has to compete for disk access with multiple front-end processes such as **smtpd**. A sudden burst of inbound mail can negatively impact outbound delivery rates.

CONFIGURATION PARAMETERS
The following **main.cf** parameters are especially relevant to this program. See the Postfix **main.cf** file for syntax details and for default values. Use the **postfix reload** command after a configuration change.

Miscellaneous
allow_min_user
Do not bounce recipient addresses that begin with '-'.

queue_directory
Top-level directory of the Postfix queue.

Active queue controls
qmgr_clog_warn_time
Minimal delay between warnings that a specific destination is clogging up the active queue. Specify 0 to disable.

qmgr_message_active_limit
Limit the number of messages in the active queue.

qmgr_message_recipient_limit
Limit the number of in-memory recipients.

This parameter also limits the size of the short-term, in-memory destination cache.

Timing controls

minimal_backoff_time
Minimal time in seconds between delivery attempts of a deferred message.

This parameter also limits the time an unreachable destination is kept in the short-term, in-memory destination status cache.

maximal_backoff_time
Maximal time in seconds between delivery attempts of a deferred message.

maximal_queue_lifetime
Maximal time in days a message is queued before it is sent back as undeliverable.

queue_run_delay
Time in seconds between deferred queue scans. Queue scans do not overlap.

transport_retry_time
Time in seconds between attempts to contact a broken delivery transport.

Concurrency controls

In the text below, *transport* is the first field in a **master.cf** entry.

qmgr_fudge_factor (valid range: 10..100)
The percentage of delivery resources that a busy mail system will use up for delivery of a large mailing list message. With 100%, delivery of one message does not begin before the previous message has been delivered. This results in good performance for large mailing lists, but results in poor response time for one-to-one mail. With less than 100%, response time for one-to-one mail improves, but large mailing list delivery performance suffers. In the worst case, recipients near the beginning of a large list receive a burst of messages immediately, while recipients near the end of that list receive that same burst of messages a whole day later.

initial_destination_concurrency
Initial per-destination concurrency level for parallel delivery to the same destination.

default_destination_concurrency_limit
Default limit on the number of parallel deliveries to the same destination.

*transport*_**destination_concurrency_limit**
Limit on the number of parallel deliveries to the same destination, for delivery via the named message *transport*.

Recipient controls

default_destination_recipient_limit
Default limit on the number of recipients per mes-
sage transfer.

*transport*_**destination_recipient_limit**
Limit on the number of recipients per message
transfer, for the named message *transport*.

SEE ALSO

master(8), process manager
syslogd(8) system logging
trivial-rewrite(8), address routing

LOCAL(8) LOCAL(8)

NAME

local - Postfix local mail delivery

SYNOPSIS

local [generic Postfix daemon options]

DESCRIPTION

The **local** daemon processes delivery requests from the Postfix queue manager to deliver mail to local recipients. Each delivery request specifies a queue file, a sender address, a domain or host to deliver to, and one or more recipients. This program expects to be run from the **master**(8) process manager.

The **local** daemon updates queue files and marks recipients as finished, or it informs the queue manager that delivery should be tried again at a later time. Delivery problem reports are sent to the **bounce**(8) or **defer**(8) daemon as appropriate.

SYSTEM-WIDE AND USER-LEVEL ALIASING

The system administrator can set up one or more system-wide **sendmail**-style alias databases. Users can have **sendmail**-style ~/.**forward** files. Mail for *name* is delivered to the alias *name*, to destinations in ~*name*/.**forward**, to the mailbox owned by the user *name*, or it is sent back as undeliverable.

The system administrator can specify a comma/space separated list of ~/.**forward** like files through the **forward_path** configuration parameter. Upon delivery, the local delivery agent tries each pathname in the list until a file is found. The **forward_path** parameter is subject to interpolation of **$user** (recipient username), **$home** (recipient home directory), **$shell** (recipient shell), **$recipient** (complete recipient address), **$extension** (recipient address extension), **$domain** (recipient domain), **local** (entire recipient address localpart) and **$recipient_delimiter**. The forms *${name?value}* and *${name:value}* expand conditionally to *value* when *$name* is (is not) defined. Characters that may have special meaning to the shell or file system are replaced by underscores. The list of acceptable characters is specified with the **forward_expansion_filter** configuration parameter.

An alias or ~/.**forward** file may list any combination of external commands, destination file names, **:include:** directives, or mail addresses. See **aliases**(5) for a precise description. Each line in a user's **.forward** file has the same syntax as the right-hand part of an alias.

When an address is found in its own alias expansion, delivery is made to the user instead. When a user is listed in the user's own ~/.**forward** file, delivery is made to the user's mailbox instead. An empty ~/.**forward** file means do not forward mail.

In order to prevent the mail system from using up unreasonable amounts of memory, input records read from **:include:** or from ~/.**forward** files are broken up into

chunks of length **line_length_limit**.
While expanding aliases, ~/.**forward** files, and so on, the program attempts to avoid duplicate deliveries. The **duplicate_filter_limit** configuration parameter limits the number of remembered recipients.

MAIL FORWARDING

For the sake of reliability, forwarded mail is re-submitted as a new message, so that each recipient has a separate on-file delivery status record.

In order to stop mail forwarding loops early, the software adds an optional **Delivered-To:** header with the envelope recipient address. If mail arrives for a recipient that is already listed in a **Delivered-To:** header, the message is bounced.

MAILBOX DELIVERY

The default per-user mailbox is a file in the UNIX mail spool directory (**/var/mail/***user* or **/var/spool/mail/***user*); the location can be specified with the **mail_spool_directory** configuration parameter. Specify a name ending in **/** for **qmail**-compatible **maildir** delivery.

Alternatively, the per-user mailbox can be a file in the user's home directory with a name specified via the **home_mailbox** configuration parameter. Specify a relative path name. Specify a name ending in **/** for **qmail**-compatible **maildir** delivery.

Mailbox delivery can be delegated to an external command specified with the **mailbox_command** configuration parameter. The command executes with the privileges of the recipient user (exception: in case of delivery as root, the command executes with the privileges of **default_privs**).

Mailbox delivery can be delegated to alternative message transports specified in the **master.cf** file. The **mailbox_transport** configuration parameter specifies a message transport that is to be used for all local recipients, regardless of whether they are found in the UNIX passwd database. The **fallback_transport** parameter specifies a message transport for recipients that are not found in the UNIX passwd database.

In the case of UNIX-style mailbox delivery, the **local** daemon prepends a "**From** *sender time_stamp*" envelope header to each message, prepends an **X-Original-To:** header with the recipient address as given to Postfix, prepends an optional **Delivered-To:** header with the envelope recipient address, prepends a **Return-Path:** header with the envelope sender address, prepends a > character to lines beginning with "**From** ", and appends an empty line. The mailbox is locked for exclusive access while delivery is in progress. In case of problems, an attempt is made to truncate the mailbox to its original length.

In the case of **maildir** delivery, the local daemon prepends an optional **Delivered-To:** header with the final envelope recipient address, prepends an **X-Original-To:** header with the recipient address as given to Postfix, and prepends a

Return-Path: header with the envelope sender address.

EXTERNAL COMMAND DELIVERY

The **allow_mail_to_commands** configuration parameter restricts delivery to external commands. The default setting (**alias, forward**) forbids command destinations in **:include:** files.

The command is executed directly where possible. Assistance by the shell (**/bin/sh** on UNIX systems) is used only when the command contains shell magic characters, or when the command invokes a shell built-in command.

A limited amount of command output (standard output and standard error) is captured for inclusion with non-delivery status reports. A command is forcibly terminated if it does not complete within **command_time_limit** seconds. Command exit status codes are expected to follow the conventions defined in <**sysexits.h**>.

A limited amount of message context is exported via environment variables. Characters that may have special meaning to the shell are replaced by underscores. The list of acceptable characters is specified with the **command_expansion_filter** configuration parameter.

SHELL The recipient user's login shell.

HOME The recipient user's home directory.

USER The bare recipient name.

EXTENSION
The optional recipient address extension.

DOMAIN The recipient address domain part.

LOGNAME
The bare recipient name.

LOCAL The entire recipient address localpart (text to the left of the rightmost @ character).

RECIPIENT
The entire recipient address.

SENDER The entire sender address.

The **PATH** environment variable is always reset to a system-dependent default path, and environment variables whose names are blessed by the **export_environment** configuration parameter are exported unchanged.

The current working directory is the mail queue directory.

The **local** daemon prepends a "**From** *sender time_stamp*" envelope header to each message, prepends an **X-Original-To:** header with the recipient address as given to Postfix, prepends an optional **Delivered-To:** header with the recipient envelope address, prepends a **Return-Path:** header with the sender envelope address, and appends no empty line.

EXTERNAL FILE DELIVERY

The delivery format depends on the destination filename

syntax. The default is to use UNIX-style mailbox format. Specify a name ending in **/** for **qmail**-compatible **maildir** delivery.

The **allow_mail_to_files** configuration parameter restricts delivery to external files. The default setting (**alias, forward**) forbids file destinations in **:include:** files.

In the case of UNIX-style mailbox delivery, the **local** daemon prepends a "**From** *sender time_stamp*" envelope header to each message, prepends an **X-Original-To:** header with the recipient address as given to Postfix, prepends an optional **Delivered-To:** header with the recipient envelope address, prepends a > character to lines beginning with "**From** ", and appends an empty line. The envelope sender address is available in the **Return-Path:** header. When the destination is a regular file, it is locked for exclusive access while delivery is in progress. In case of problems, an attempt is made to truncate a regular file to its original length.

In the case of **maildir** delivery, the local daemon prepends an optional **Delivered-To:** header with the envelope recipient address, and prepends an **X-Original-To:** header with the recipient address as given to Postfix. The envelope sender address is available in the **Return-Path:** header.

ADDRESS EXTENSION

The optional **recipient_delimiter** configuration parameter specifies how to separate address extensions from local recipient names.

For example, with "**recipient_delimiter = +**", mail for *name*+*foo* is delivered to the alias *name*+*foo* or to the alias *name*, to the destinations listed in ~*name*/.**forward**+*foo* or in ~*name*/.**forward**, to the mailbox owned by the user *name*, or it is sent back as undeliverable.

In all cases the **local** daemon prepends an optional `**Delivered-To:** *name*+*foo*' header line.

DELIVERY RIGHTS

Deliveries to external files and external commands are made with the rights of the receiving user on whose behalf the delivery is made. In the absence of a user context, the **local** daemon uses the owner rights of the **:include:** file or alias database. When those files are owned by the superuser, delivery is made with the rights specified with the **default_privs** configuration parameter.

STANDARDS

RFC 822 (ARPA Internet Text Messages)

DIAGNOSTICS

Problems and transactions are logged to **syslogd**(8). Corrupted message files are marked so that the queue manager can move them to the **corrupt** queue afterwards.

Depending on the setting of the **notify_classes** parameter, the postmaster is notified of bounces and of other trouble.

BUGS

For security reasons, the message delivery status of external commands or of external files is never checkpointed to file. As a result, the program may occasionally deliver more than once to a command or external file. Better safe than sorry.

Mutually-recursive aliases or ~/.**forward** files are not detected early. The resulting mail forwarding loop is broken by the use of the **Delivered-To:** message header.

CONFIGURATION PARAMETERS

The following **main.cf** parameters are especially relevant to this program. See the Postfix **main.cf** file for syntax details and for default values. Use the **postfix reload** command after a configuration change.

Miscellaneous

alias_maps
List of alias databases.

biff Enable or disable notification of new mail via the **comsat** network service.

expand_owner_alias
When delivering to an alias that has an owner- companion alias, set the envelope sender address to the right-hand side of the owner alias, instead using of the left-hand side address.

export_environment
List of names of environment parameters that can be exported to non-Postfix processes.

forward_path
Search list for .forward files. The names are subject to *$name* expansion.

local_command_shell
Shell to use for external command execution (for example, /some/where/smrsh -c). When a shell is specified, it is invoked even when the command contains no shell built-in commands or meta characters.

owner_request_special
Give special treatment to **owner-***xxx* and *xxx*-**request** addresses.

prepend_delivered_header
Prepend an optional **Delivered-To:** header upon external forwarding, delivery to command or file. Specify zero or more of: **command**, **file**, **forward**. Turning off **Delivered-To:** when forwarding mail is not recommended.

recipient_delimiter
Separator between username and address extension.

require_home_directory
Require that a recipient's home directory is accessible by the recipient before attempting delivery.

Defer delivery otherwise.

Mailbox delivery

fallback_transport
Message transport for recipients that are not found in the UNIX passwd database. This parameter overrides **luser_relay**.

Note: you must update the **local_recipient_maps** setting in the **main.cf** file, otherwise the Postfix SMTP server will reject mail for non-UNIX accounts with **"User unknown in local recipient table"**.

home_mailbox
Pathname of a mailbox relative to a user's home directory. Specify a path ending in **/** for maildir-style delivery.

luser_relay
Destination (*@domain* or *address*) for non-existent users. The *address* is subjected to *$name* expansion.

Note: you must specify **"local_recipient_maps ="** (i.e. empty) in the **main.cf** file, otherwise the Postfix SMTP server will reject mail for non-UNIX accounts with **"User unknown in local recipient table"**.

mail_spool_directory
Directory with UNIX-style mailboxes. The default pathname is system dependent. Specify a path ending in **/** for maildir-style delivery.

mailbox_command
External command to use for mailbox delivery. The command executes with the recipient privileges (exception: root). The string is subject to $name expansions.

mailbox_command_maps
Lookup tables with per-recipient external commands to use for mailbox delivery. Behavior is as with **mailbox_command**.

mailbox_transport
Message transport to use for mailbox delivery to all local recipients, whether or not they are found in the UNIX passwd database. This parameter overrides all other configuration parameters that control mailbox delivery, including **luser_relay**.

Note: if you use this feature to receive mail for non-UNIX accounts then you must update the **local_recipient_maps** setting in the **main.cf** file, otherwise the Postfix SMTP server will reject mail for non-UNIX accounts with **"User unknown in local recipient table"**.

Locking controls

deliver_lock_attempts
Limit the number of attempts to acquire an exclu-

sive lock on a mailbox or external file.

deliver_lock_delay
Time in seconds between successive attempts to acquire an exclusive lock.

stale_lock_time
Limit the time after which a stale lock is removed.

mailbox_delivery_lock
What file locking method(s) to use when delivering to a UNIX-style mailbox. The default setting is system dependent. For a list of available file locking methods, use the **postconf -l** command.

Resource controls

command_time_limit
Limit the amount of time for delivery to external command.

duplicate_filter_limit
Limit the size of the duplicate filter for results from alias etc. expansion.

line_length_limit
Limit the amount of memory used for processing a partial input line.

local_destination_concurrency_limit
Limit the number of parallel deliveries to the same user. The default limit is taken from the **default_destination_concurrency_limit** parameter.

local_destination_recipient_limit
Limit the number of recipients per message delivery. The default limit is taken from the **default_destination_recipient_limit** parameter.

mailbox_size_limit
Limit the size of a mailbox etc. file (any file that is written to upon delivery). Set to zero to disable the limit.

Security controls

allow_mail_to_commands
Restrict the usage of mail delivery to external command. Specify zero or more of: **alias, forward, include.**

allow_mail_to_files
Restrict the usage of mail delivery to external file. Specify zero or more of: **alias, forward, include.**

command_expansion_filter
What characters are allowed to appear in $name expansions of mailbox_command. Illegal characters are replaced by underscores.

default_privs
Default rights for delivery to external file or command.

forward_expansion_filter
What characters are allowed to appear in $name expansions of forward_path. Illegal characters are replaced by underscores.

HISTORY

The **Delivered-To:** header appears in the **qmail** system by Daniel Bernstein.

The *maildir* structure appears in the **qmail** system by Daniel Bernstein.

SMTP(8) SMTP(8)

NAME

smtp - Postfix remote delivery via SMTP

SYNOPSIS

smtp [generic Postfix daemon options]

DESCRIPTION

The SMTP client processes message delivery requests from the queue manager. Each request specifies a queue file, a sender address, a domain or host to deliver to, and recipient information. This program expects to be run from the **master**(8) process manager.

The SMTP client updates the queue file and marks recipients as finished, or it informs the queue manager that delivery should be tried again at a later time. Delivery problem reports are sent to the **bounce**(8) or **defer**(8) daemon as appropriate.

The SMTP client looks up a list of mail exchanger addresses for the destination host, sorts the list by preference, and connects to each listed address until it finds a server that responds.

When the domain or host is specified as a comma/whitespace separated list, the SMTP client repeats the above process for all destinations until it finds a server that responds.

Once the SMTP client has received the server greeting banner, no error will cause it to proceed to the next address on the mail exchanger list. Instead, the message is either bounced, or its delivery is deferred until later.

SECURITY

The SMTP client is moderately security-sensitive. It talks to SMTP servers and to DNS servers on the network. The SMTP client can be run chrooted at fixed low privilege.

STANDARDS

RFC 821 (SMTP protocol)
RFC 822 (ARPA Internet Text Messages)
RFC 1651 (SMTP service extensions)
RFC 1652 (8bit-MIME transport)
RFC 1870 (Message Size Declaration)
RFC 2045 (MIME: Format of Internet Message Bodies)
RFC 2046 (MIME: Media Types)
RFC 2554 (AUTH command)
RFC 2821 (SMTP protocol)
RFC 2920 (SMTP Pipelining)

DIAGNOSTICS

Problems and transactions are logged to **syslogd**(8). Corrupted message files are marked so that the queue manager can move them to the **corrupt** queue for further inspection.

Depending on the setting of the **notify_classes** parameter, the postmaster is notified of bounces, protocol problems,

```
       and of other trouble.
```

BUGS

CONFIGURATION PARAMETERS

The following **main.cf** parameters are especially relevant to this program. See the Postfix **main.cf** file for syntax details and for default values. Use the **postfix reload** command after a configuration change.

Miscellaneous

best_mx_transport
Name of the delivery transport to use when the local machine is the most-preferred mail exchanger (by default, a mailer loop is reported, and the message is bounced).

debug_peer_level
Verbose logging level increment for hosts that match a pattern in the **debug_peer_list** parameter.

debug_peer_list
List of domain or network patterns. When a remote host matches a pattern, increase the verbose logging level by the amount specified in the **debug_peer_level** parameter.

disable_dns_lookups
Disable DNS lookups. This means that mail must be forwarded via a smart relay host.

error_notice_recipient
Recipient of protocol/policy/resource/software error notices.

fallback_relay
Hosts to hand off mail to if a message destination is not found or if a destination is unreachable.

ignore_mx_lookup_error
When a name server fails to respond to an MX query, search for an A record instead deferring mail delivery.

inet_interfaces
The network interface addresses that this mail system receives mail on. When any of those addresses appears in the list of mail exchangers for a remote destination, the list is truncated to avoid mail delivery loops. See also the **proxy_interfaces** parameter.

notify_classes
When this parameter includes the **protocol** class, send mail to the postmaster with transcripts of SMTP sessions with protocol errors.

proxy_interfaces
Network interfaces that this mail system receives mail on by way of a proxy or network address translator. When any of those addresses appears in the list of mail exchangers for a remote destination, the list is truncated to avoid mail delivery loops. See also the **inet_interfaces** parameter.

smtp_always_send_ehlo

```
Always send EHLO at the start of a connection.
```

smtp_never_send_ehlo
Never send EHLO at the start of a connection.

smtp_bind_address
Numerical source network address to bind to when making a connection.

smtp_line_length_limit
Length limit for SMTP message content lines. Zero means no limit. Some SMTP servers misbehave on long lines.

smtp_helo_name
The hostname to be used in HELO and EHLO commands.

smtp_skip_4xx_greeting
Skip servers that greet us with a 4xx status code.

smtp_skip_5xx_greeting
Skip servers that greet us with a 5xx status code.

smtp_skip_quit_response
Do not wait for the server response after sending QUIT.

smtp_pix_workaround_delay_time
The time to pause before sending .<CR><LF>, while working around the CISCO PIX firewall <CR><LF>.<CR><LF> bug.

smtp_pix_workaround_threshold_time
The time a message must be queued before the CISCO PIX firewall <CR><LF>.<CR><LF> bug workaround is turned on.

MIME Conversion

disable_mime_output_conversion
Disable the conversion of 8BITMIME format to 7BIT format when the remote system does not advertise 8BITMIME support.

mime_boundary_length_limit
The amount of space that will be allocated for MIME multipart boundary strings. The MIME processor is unable to distinguish between boundary strings that do not differ in the first **$mime_boundary_length_limit** characters.

mime_nesting_limit
The maximal nesting level of multipart mail that the MIME processor can handle. Refuse mail that is nested deeper, when converting from 8BITMIME format to 7BIT format.

Authentication controls

smtp_sasl_auth_enable
Enable per-session authentication as per RFC 2554 (SASL). By default, Postfix is built without SASL support.

smtp_sasl_password_maps
Lookup tables with per-host or domain *name*:*password* entries. No entry for a host means no attempt to

```
authenticate.
```

smtp_sasl_security_options
Zero or more of the following.

noplaintext
Disallow authentication methods that use plaintext passwords.

noactive
Disallow authentication methods that are vulnerable to non-dictionary active attacks.

nodictionary
Disallow authentication methods that are vulnerable to passive dictionary attack.

noanonymous
Disallow anonymous logins.

Resource controls

smtp_destination_concurrency_limit
Limit the number of parallel deliveries to the same destination. The default limit is taken from the **default_destination_concurrency_limit** parameter.

smtp_destination_recipient_limit
Limit the number of recipients per message delivery. The default limit is taken from the **default_destination_recipient_limit** parameter.

Timeout controls

The default time unit is seconds; an explicit time unit can be specified by appending a one-letter suffix to the value: s (seconds), m (minutes), h (hours), d (days) or w (weeks).

smtp_connect_timeout
Timeout for completing a TCP connection. When no connection can be made within the deadline, the SMTP client tries the next address on the mail exchanger list.

smtp_helo_timeout
Timeout for receiving the SMTP greeting banner. When the server drops the connection without sending a greeting banner, or when it sends no greeting banner within the deadline, the SMTP client tries the next address on the mail exchanger list.

smtp_helo_timeout
Timeout for sending the **HELO** command, and for receiving the server response.

smtp_mail_timeout
Timeout for sending the **MAIL FROM** command, and for receiving the server response.

smtp_rcpt_timeout
Timeout for sending the **RCPT TO** command, and for receiving the server response.

smtp_data_init_timeout
Timeout for sending the **DATA** command, and for receiving the server response.

smtp_data_xfer_timeout
Timeout for sending the message content.

smtp_data_done_timeout
Timeout for sending the "." command, and for receiving the server response. When no response is received, a warning is logged that the mail may be delivered multiple times.

smtp_quit_timeout
Timeout for sending the **QUIT** command, and for receiving the server response.

SEE ALSO

bounce(8) non-delivery status reports
master(8) process manager
qmgr(8) queue manager
syslogd(8) system logging

LMTP(8) LMTP(8)

NAME

lmtp - Postfix local delivery via LMTP

SYNOPSIS

lmtp [generic Postfix daemon options]

DESCRIPTION

The LMTP client processes message delivery requests from the queue manager. Each request specifies a queue file, a sender address, a domain or host to deliver to, and recipient information. This program expects to be run from the **master**(8) process manager.

The LMTP client updates the queue file and marks recipients as finished, or it informs the queue manager that delivery should be tried again at a later time. Delivery problem reports are sent to the **bounce**(8) or **defer**(8) daemon as appropriate.

The LMTP client connects to the destination specified in the message delivery request. The destination, usually specified in the Postfix **transport**(5) table, has the form:

unix:*pathname*
Connect to the local UNIX-domain server that is bound to the specified *pathname*. If the process runs chrooted, an absolute pathname is interpreted relative to the changed root directory.

inet:*host*, **inet**:*host*:*port* (symbolic host)

inet:[*addr*], **inet**:[*addr*]:*port* (numeric host)
Connect to the specified IPV4 TCP port on the specified local or remote host. If no port is specified, connect to the port defined as **lmtp** in **services**(4). If no such service is found, the **lmtp_tcp_port** configuration parameter (default value of 24) will be used.

The LMTP client does not perform MX (mail exchanger) lookups since those are defined only for mail delivery via SMTP.

If neither **unix**: nor **inet**: are specified, **inet**: is assumed.

SECURITY

The LMTP client is moderately security-sensitive. It talks to LMTP servers and to DNS servers on the network. The LMTP client can be run chrooted at fixed low privilege.

STANDARDS

RFC 821 (SMTP protocol)
RFC 1651 (SMTP service extensions)
RFC 1652 (8bit-MIME transport)
RFC 1870 (Message Size Declaration)
RFC 2033 (LMTP protocol)
RFC 2554 (AUTH command)
RFC 2821 (SMTP protocol)
RFC 2920 (SMTP Pipelining)

DIAGNOSTICS

Problems and transactions are logged to **syslogd**(8). Corrupted message files are marked so that the queue manager can move them to the **corrupt** queue for further inspection.

Depending on the setting of the **notify_classes** parameter, the postmaster is notified of bounces, protocol problems, and of other trouble.

BUGS

CONFIGURATION PARAMETERS

The following **main.cf** parameters are especially relevant to this program. See the Postfix **main.cf** file for syntax details and for default values. Use the **postfix reload** command after a configuration change.

Miscellaneous

debug_peer_level

Verbose logging level increment for hosts that match a pattern in the **debug_peer_list** parameter.

debug_peer_list

List of domain or network patterns. When a remote host matches a pattern, increase the verbose logging level by the amount specified in the **debug_peer_level** parameter.

error_notice_recipient

Recipient of protocol/policy/resource/software error notices.

notify_classes

When this parameter includes the **protocol** class, send mail to the postmaster with transcripts of LMTP sessions with protocol errors.

lmtp_skip_quit_response

Do not wait for the server response after sending QUIT.

lmtp_tcp_port

The TCP port to be used when connecting to a LMTP server. Used as backup if the **lmtp** service is not found in **services**(4).

Authentication controls

lmtp_sasl_auth_enable

Enable per-session authentication as per RFC 2554 (SASL). By default, Postfix is built without SASL support.

lmtp_sasl_password_maps

Lookup tables with per-host or domain *name*:*password* entries. No entry for a host means no attempt to authenticate.

lmtp_sasl_security_options

Zero or more of the following.

noplaintext

Disallow authentication methods that use

plaintext passwords.

noactive
Disallow authentication methods that are vulnerable to non-dictionary active attacks.

nodictionary
Disallow authentication methods that are vulnerable to passive dictionary attack.

noanonymous
Disallow anonymous logins.

Resource controls

lmtp_cache_connection
Should we cache the connection to the LMTP server? The effectiveness of cached connections will be determined by the number of LMTP servers in use, and the concurrency limit specified for the LMTP client. Cached connections are closed under any of the following conditions:

- The LMTP client idle time limit is reached. This limit is specified with the Postfix **max_idle** configuration parameter.
- A delivery request specifies a different destination than the one currently cached.
- The per-process limit on the number of delivery requests is reached. This limit is specified with the Postfix **max_use** configuration parameter.
- Upon the onset of another delivery request, the LMTP server associated with the current session does not respond to the **RSET** command.

*transport*_**destination_concurrency_limit**
Limit the number of parallel deliveries to the same destination via this mail delivery transport. *transport* is the name of the service as specified in the **master.cf** file. The default limit is taken from the **default_destination_concurrency_limit** parameter.

*transport*_**destination_recipient_limit**
Limit the number of recipients per message delivery via this mail delivery transport. *transport* is the name of the service as specified in the **master.cf** file. The default limit is taken from the **default_destination_recipient_limit** parameter.

This parameter becomes significant if the LMTP client is used for local delivery. Some LMTP servers can optimize delivery of the same message to multiple recipients. The default limit for local mail delivery is 1.

Setting this parameter to 0 will lead to an unbounded number of recipients per delivery. However, this could be risky since it may make the machine vulnerable to running out of resources if

```
messages are encountered with an inordinate number
of recipients.  Exercise care when setting this
parameter.
```

Timeout controls

The default time unit is seconds; an explicit time unit can be specified by appending a one-letter suffix to the value: s (seconds), m (minutes), h (hours), d (days) or w (weeks).

lmtp_connect_timeout
Timeout for opening a connection to the LMTP server. If no connection can be made within the deadline, the message is deferred.

lmtp_lhlo_timeout
Timeout for sending the **LHLO** command, and for receiving the server response.

lmtp_mail_timeout
Timeout for sending the **MAIL FROM** command, and for receiving the server response.

lmtp_rcpt_timeout
Timeout for sending the **RCPT TO** command, and for receiving the server response.

lmtp_data_init_timeout
Timeout for sending the **DATA** command, and for receiving the server response.

lmtp_data_xfer_timeout
Timeout for sending the message content.

lmtp_data_done_timeout
Timeout for sending the "." command, and for receiving the server response. When no response is received, a warning is logged that the mail may be delivered multiple times.

lmtp_rset_timeout
Timeout for sending the **RSET** command, and for receiving the server response.

lmtp_quit_timeout
Timeout for sending the **QUIT** command, and for receiving the server response.

SEE ALSO

bounce(8) non-delivery status reports
local(8) local mail delivery
master(8) process manager
qmgr(8) queue manager
services(4) Internet services and aliases
spawn(8) auxiliary command spawner
syslogd(8) system logging

LICENSE

The Secure Mailer license must be distributed with this software.

AUTHOR(S)

Wietse Venema

PIPE(8) PIPE(8)

NAME

pipe - Postfix delivery to external command

SYNOPSIS

pipe [generic Postfix daemon options] command_attributes...

DESCRIPTION

The **pipe** daemon processes requests from the Postfix queue manager to deliver messages to external commands. This program expects to be run from the **master**(8) process manager.

Message attributes such as sender address, recipient address and next-hop host name can be specified as command-line macros that are expanded before the external command is executed.

The **pipe** daemon updates queue files and marks recipients as finished, or it informs the queue manager that delivery should be tried again at a later time. Delivery problem reports are sent to the **bounce**(8) or **defer**(8) daemon as appropriate.

SINGLE-RECIPIENT DELIVERY

Some external commands cannot handle more than one recipient per delivery request. Examples of such transports are pagers, fax machines, and so on.

To prevent Postfix from sending multiple recipients per delivery request, specify

*transport*_**destination_recipient_limit = 1**

in the Postfix **main.cf** file, where *transport* is the name in the first column of the Postfix **master.cf** entry for the pipe-based delivery transport.

COMMAND ATTRIBUTE SYNTAX

The external command attributes are given in the **master.cf** file at the end of a service definition. The syntax is as follows:

flags=BDFORhqu.> (optional)

Optional message processing flags. By default, a message is copied unchanged.

B Append a blank line at the end of each message. This is required by some mail user agents that recognize **"From "** lines only when preceded by a blank line.

D Prepend a **"Delivered-To:** *recipient*" message header with the envelope recipient address. Note: for this to work, the *transport*_**destination_recipient_limit** must be 1.

F Prepend a **"From** *sender time_stamp*" envelope header to the message content. This is expected by, for example, **UUCP** software.

O Prepend an "**X-Original-To:** *recipient*" message header with the recipient address as given to Postfix. Note: for this to work, the *transport*_**destination_recipient_limit** must be 1.

R Prepend a **Return-Path:** message header with the envelope sender address.

h Fold the command-line **$recipient** domain name and **$nexthop** host name to lower case. This is recommended for delivery via **UUCP**.

q Quote white space and other special characters in the command-line **$sender** and **$recipient** address localparts (text to the left of the right-most **@** character), according to an 8-bit transparent version of RFC 822. This is recommended for delivery via **UUCP** or **BSMTP**.

The result is compatible with the address parsing of command-line recipients by the Postfix **sendmail** mail submission command.

The **q** flag affects only entire addresses, not the partial address information from the **$user**, **$extension** or **$mailbox** command-line macros.

u Fold the command-line **$recipient** address localpart (text to the left of the right-most **@** character) to lower case. This is recommended for delivery via **UUCP**.

. Prepend . to lines starting with ".". This is needed by, for example, **BSMTP** software.

> Prepend > to lines starting with "**From** ". This is expected by, for example, **UUCP** software.

user=*username* (required)

user=*username*:*groupname*
The external command is executed with the rights of the specified *username*. The software refuses to execute commands with root privileges, or with the privileges of the mail system owner. If *groupname* is specified, the corresponding group ID is used instead of the group ID of *username*.

eol=string (optional, default: **\n**)
The output record delimiter. Typically one would use either **\r\n** or **\n**. The usual C-style backslash escape sequences are recognized: **\a \b \f \n \r \t \v **_octal_ and ****.

size=*size_limit* (optional)
Messages greater in size than this limit (in bytes) will be bounced back to the sender.

argv=*command*... (required)
The command to be executed. This must be specified as the last command attribute. The command is executed directly, i.e. without interpretation of shell meta characters by a shell command interpreter.

In the command argument vector, the following macros are recognized and replaced with corresponding information from the Postfix queue manager delivery request:

${extension}
This macro expands to the extension part of a recipient address. For example, with an address *user+foo@domain* the extension is *foo*.

A command-line argument that contains **${extension**} expands into as many command-line arguments as there are recipients.

This information is modified by the **u** flag for case folding.

${mailbox}
This macro expands to the complete local part of a recipient address. For example, with an address *user+foo@domain* the mailbox is *user+foo*.

A command-line argument that contains **${mailbox**} expands into as many command-line arguments as there are recipients.

This information is modified by the **u** flag for case folding.

${nexthop}
This macro expands to the next-hop hostname.

This information is modified by the **h** flag for case folding.

${recipient}
This macro expands to the complete recipient address.

A command-line argument that contains **${recipient**} expands into as many command-line arguments as there are recipients.

This information is modified by the **hqu** flags for quoting and case folding.

${sender}
This macro expands to the envelope sender address.

This information is modified by the **q** flag for quoting.

${size}
This macro expands to Postfix's idea of the

message size, which is an approximation of the size of the message as delivered.

${user}
This macro expands to the username part of a recipient address. For example, with an address *user+foo@domain* the username part is *user*.

A command-line argument that contains **${user**} expands into as many command-line arguments as there are recipients.

This information is modified by the **u** flag for case folding.

In addition to the form ${*name*}, the forms $*name* and $(*name*) are also recognized. Specify **$$** where a single **$** is wanted.

DIAGNOSTICS

Command exit status codes are expected to follow the conventions defined in <**sysexits.h**>.

Problems and transactions are logged to **syslogd**(8). Corrupted message files are marked so that the queue manager can move them to the **corrupt** queue for further inspection.

SECURITY

This program needs a dual personality 1) to access the private Postfix queue and IPC mechanisms, and 2) to execute external commands as the specified user. It is therefore security sensitive.

CONFIGURATION PARAMETERS

The following **main.cf** parameters are especially relevant to this program. See the Postfix **main.cf** file for syntax details and for default values. Use the **postfix reload** command after a configuration change.

Miscellaneous

export_environment
List of names of environment parameters that can be exported to non-Postfix processes.

mail_owner
The process privileges used while not running an external command.

Resource controls

In the text below, *transport* is the first field in a **master.cf** entry.

*transport*_**destination_concurrency_limit**
Limit the number of parallel deliveries to the same destination, for delivery via the named *transport*. The default limit is taken from the **default_destination_concurrency_limit** parameter. The limit is enforced by the Postfix queue manager.

*transport*_**destination_recipient_limit**
Limit the number of recipients per message deliv-

ery, for delivery via the named *transport*. The default limit is taken from the **default_destination_recipient_limit** parameter. The limit is enforced by the Postfix queue manager.

*transport*_**time_limit**
Limit the time for delivery to external command, for delivery via the named **transport**. The default limit is taken from the **command_time_limit** parameter. The limit is enforced by the pipe delivery agent.

SEE ALSO

bounce(8) non-delivery status reports
master(8) process manager
qmgr(8) queue manager
syslogd(8) system logging

LICENSE

The Secure Mailer license must be distributed with this software.

What domain to use in outbound mail

The myorigin parameter specifies the domain that appears in mail that is posted on this machine. The default is to use the local machine name, `$myhostname`, which defaults to the name of the machine. Unless you are running a really small site, you probably want to change that into `$mydomain`, which defaults to the parent domain of the machine name.

For the sake of consistency between sender and recipient addresses, `myorigin` also specifies the default domain name that is appended to an unqualified recipient address.

Examples:

```
myorigin = $myhostname (default)
myorigin = $mydomain   (probably desirable)
```

What domains to receive mail for

The `mydestination` parameter specifies what domains this machine will deliver locally, instead of forwarding to another machine. The default is to receive mail for the machine itself.

You can specify zero or more domain names, */file/name* patterns and/or *type:name* lookup tables, separated by whitespace and/or commas. A */file/name* is replaced by its contents; *type:name* requests that a table lookup is done. If your machine is a mail server for its entire domain, you must list `$mydomain` as well.

Examples:

Default setting:

```
mydestination = $myhostname localhost.$mydomain
```

Domain-wide mail server:

```
mydestination = $myhostname localhost.$mydomain $mydomain
```

Host with multiple DNS A records:

```
mydestination = $myhostname localhost.$mydomain www.$mydomain
                    ftp.$mydomain
```

Caution: in order to avoid mail delivery loops, you must list all hostnames of the machine, including $myhostname, and localhost.$mydomain.

What clients to relay mail for

By default, Postfix will relay mail for clients in authorized networks.

Authorized client networks are defined by the `mynetworks` parameter. The default is to authorize all clients in the IP subnetworks that the local machine is attached to.

What trouble to report to the postmaster

You should set up a **`postmaster`** `alias` that points to a human person. This alias is required to exist, so that people can report mail delivery problems.

The Postfix system itself also reports problems to the postmaster alias. You may not be interested in all types of trouble reports, so this reporting mechanism is configurable. The default is to report only serious problems (resource, software) to postmaster:

Default:

```
notify_classes = resource, software
```

The meaning of the classes is as follows:

bounce
Send postmaster copies of undeliverable mail. If mail is undeliverable, a so-called single bounce message is sent, with a copy of the message that was not delivered. For privacy reasons, the postmaster copy of a single bounce message is truncated after the original message headers. If a single bounce message is undeliverable, the postmaster receives a double bounce message with a copy of the entire single bounce message. See also the `luser_relay` feature.

2bounce
Send double bounces to the postmaster.

delay
Inform the postmaster of delayed mail. In this case, the postmaster receives message headers only.

policy
Inform the postmaster of client requests that were rejected because of (UCE) policy restrictions. The postmaster receives a transcript of the entire SMTP session.

protocol
Inform the postmaster of protocol errors (client or server side) or attempts by a client to execute unimplemented commands. The postmaster receives a transcript of the entire SMTP session.

resource
Inform the postmaster of mail not delivered due to resource problems (for example, queue file write errors).

software
Inform the postmaster of mail not delivered due to software problems.

Proxy/NAT network addresses

The `proxy_interfaces` parameter specifies all network addresses that the Postfix receives mail on by way of a proxy or network address translation unit. You may specify symbolic hostnames instead of network addresses.

You must specify your proxy/NAT addresses when your system is a backup **MX** host for other domains, otherwise mail delivery loops will happen when the primary **MX** host is down.

Examples:

Default:
```
proxy_interfaces =
```

Host running backup MTA:
`proxy_interfaces = 1.2.3.4` (the proxy/NAT network address)

My own hostname

The `myhostname` parameter describes the fully-qualified domain name of the machine running the Postfix system. `$myhostname` appears as the default value in many other Postfix configuration parameters.

By default, `myhostname` is set to the local machine name. If your machine name is not in fully-qualified domain name form, or if you run Postfix on a virtual interface, you will have

to specify the fully-qualified domain name that the mail system should use.

Examples:

```
myhostname = host.local.domain      (local hostname is not FQDN)
myhostname = host.virtual.domain    (virtual interface)
myhostname = virtual.domain         (virtual interface)
```

My own domain name

The `mydomain` parameter specifies the parent domain of `$myhostname`**.** By default it is derived from `$myhostname` by stripping off the first part (unless the result would be a top-level domain).

Examples:

```
mydomain = local.domain
mydomain = virtual.domain (virtual interface)
```

My own networks

The `mynetworks` parameter lists all networks that this machine somehow trusts. This information can be used by the anti-UCE features to recognize trusted SMTP clients that are allowed to relay mail through Postfix.

You can specify the list of trusted networks in the **`main.cf`** file, or you can let Postfix deduce the list for you. The default is to let Postfix do the work for you.

Default:

```
mynetworks_style = subnet
```

The meaning of the styles is as follows:

`class`
Trust SMTP clients in the class A/B/C networks that Postfix is connected to. **Don't do this with a dialup site - it would cause Postfix to "trust" your entire provider's network. Instead, specify an explicit mynetworks list by hand, as described below**.

`subnet` (default)
Trust SMTP clients in the IP subnetworks that Postfix is connected to.

`host` Trust only the local machine.

Alternatively, you can specify the `mynetworks` list by hand, in which case Postfix ignores the `mynetworks_style` setting. To specify the list of trusted networks by hand, specify network blocks in CIDR (network/mask) notation, for example:

```
mynetworks = 168.100.189.0/28, 127.0.0.0/8
```

You can also specify the absolute pathname of a pattern file instead of listing the patterns in the **`main.cf`** file.

My own network addresses

The `inet_interfaces` parameter specifies all network interface addresses that the Postfix system should listen on; mail addressed to *`user`*`@[`*`network address`*`]` will be delivered locally, as if it is addressed to a domain listed in `$mydestination`**.**

The default is to listen on all active interfaces. If you run mailers on virtual interfaces, you will have to specify what interfaces to listen on.

You even have to specify explicit machine interfaces for the non-virtual mailer that receives mail for the machine itself: the non-virtual mailer should never listen on the virtual interfaces or you would have a mailer loop.

Examples:

Default:

```
inet_interfaces = all
```

Host running virtual mailers:

`inet_interfaces = virtual.host.tld` (virtual domain)

`inet_interfaces = $myhostname localhost.$mydomain` (non-virtual mailer)

Note: you need to stop and start Postfix when this parameter changes.

Postfix Configuration - UCE Controls

Introduction

Postfix offers a variety of parameters that limit the delivery of unsolicited commercial email (UCE).

By default, the Postfix SMTP server will accept mail only from or to the local network or domain, or to domains that are hosted by Postfix, so that your system can't be used as a mail relay to forward bulk mail from random strangers.

The text in this document describes how you can set up more detailed anti-UCE policies that prevent delivery of unwanted email altogether, for example with sendmail-style **access** lists or with **RBL** (real-time blackhole list) name servers.

Unless indicated otherwise, all parameters described here are in the **`main.cf`** file. If you change parameters of a running Postfix system, don't forget to issue a `postfix reload` command.

Header filtering

The `header_checks` parameter restricts what is allowed in message headers. Patterns are applied to entire logical message headers, even when a header spans multiple lines of text.

By default, the same `header_checks` patterns are used for primary message headers, for MIME headers (including headers at the start of multipart body parts), and for the headers at the beginning of attached email messages.

Default:

Allow anything in message headers.

Syntax:

Specify a list of zero or more lookup tables. Whenever a header matches a table, the action depends on the lookup result:

REJECT
REJECT *text...*
Reject the message, log the header and the optional text, and send the optional text to the originator.
IGNORE
Delete the header from the message.
WARN
WARN *text...*
Log (but do not reject) the header with a warning, and log the optional text.
HOLD
HOLD *text...*
Place the message on the **hold** queue. Mail on hold can be inspected with the postcat command, and can be destroyed or taken off hold with the postsuper command. The optional text is logged together with the matched text.
DISCARD
DISCARD *text...*
Claim successful delivery and silently discard the message. The optional text is logged together with the matched text.
FILTER *transport*:*nexthop*
After the message is queued, send the entire message through a content filter. This requires different cleanup servers before and after the filter, with header/body checks turned off in the second cleanup server. More details about content filtering are in the Postfix FILTER_README file. This feature overrides the **main.cf** `content_filter` setting.

At present, specifying a header pattern with OK serves no useful purpose. A rule ending in OK affects only the header being matched. The next header may still result in a REJECT match, causing the mail still to be rejected.

Examples (main.cf):

```
header_checks = regexp:/etc/postfix/header_checks
header_checks = pcre:/etc/postfix/header_checks
```

Example (header_checks):

```
/^to: *friend@public\.com$/ REJECT
```

Body filtering

The `body_checks` parameter restricts what text is is allowed in message body lines.

Note: the message body is matched one line at a time. There is no multi-line concept as with message headers.

Default:

Allow anything in message body lines.

Syntax:

Specify a list of zero or more lookup tables. Whenever a body line matches a table, the action depends on the lookup result:

REJECT
REJECT *text...*
Reject the message, log the body line and the optional text, and send the optional text to the originator.
WARN
WARN *text...*
Log (but do not reject) the body line with a warning, and log the optional text.
IGNORE
Delete the matched line from the message.
HOLD
HOLD *text...*
Place the message on the **hold** queue. Mail on hold can be inspected with the postcat command, and can be destroyed or taken off hold with the postsuper command. The optional text is logged together with the matched text.
DISCARD
DISCARD *text...*
Claim successful delivery and silently discard the message. The optional text is logged together with the matched text.
FILTER *transport*:*nexthop*
After the message is queued, send the entire message through a content filter. This requires different cleanup servers before and after the filter, with header/body checks turned off in the second cleanup server. More details about content filtering are in the Postfix FILTER_README file. This feature overrides the **main.cf** `content_filter` setting.

At present, specifying a pattern with OK serves no useful purpose. A rule ending in OK affects only the line being matched. The next line may still result in a REJECT match, causing the mail still to be rejected.

Examples (main.cf):

```
body_checks = regexp:/etc/postfix/body_checks
body_checks = pcre:/etc/postfix/body_checks
```

Client hostname/address restrictions

The `smtpd_client_restrictions` parameter restricts what clients this system accepts SMTP connections from.

By default, this restriction is applied when the client sends the RCPT TO command. In order to have the restriction take effect as soon as possible, specify **`smtpd_delay_reject = no`** in the Postfix **`main.cf`** configuration file. Doing so may cause unexpected results with poorly implemented client software.

Default:

```
smtpd_client_restrictions =
```

Allow SMTP connections from any client.

Syntax:

Specify a list of zero or more restrictions, separated by whitespace or commas. Restrictions are applied in the order as specified; the first restriction that matches wins.

In addition to restrictions that are specific to the client hostname or IP address, you may list here any restrictions based on the information passed with the

HELO/EHLO command, on the sender address or on the recipient address. The HELO/EHLO, sender or recipient restrictions take effect only if **smtpd_delay_reject = yes** so that all restrictions are evaluated after the RCPT TO command.

Examples:

```
smtpd_client_restrictions = hash:/etc/postfix/access,
reject_rbl_client relays.mail-abuse.org      (paid service)

smtpd_client_restrictions = hash:/etc/postfix/access,
reject_rbl_client relays.ordb.org            (free service)

smtpd_client_restrictions = hash:/etc/postfix/access,
reject_rhsbl_client dsn.rfc-ignorant.org     (free service)

smtpd_client_restrictions = permit_mynetworks,
reject_unknown_client
```

Restrictions:

`reject_unknown_client`
: Reject the request when the client IP address has no PTR (address to name) record in the DNS, or when the PTR record does not have a matching A (name to address) record. The `unknown_client_reject_code` parameter specifies the response code to rejected requests (default: **450**).

`permit_mynetworks`
: Permit the request when the client IP address matches any network listed in `$mynetworks`.

`reject_rbl_client` *`domain.tld`*
: Reject the request when the reversed client network address is listed with an A record under *`domain.tld`*. The `maps_rbl_reject_code` parameter specifies the response code for rejected requests (default: **554**), the `default_rbl_reply` parameter specifies the default server reply, and the `rbl_reply_maps` parameter specifies tables with server replies indexed by RBL domain.

`reject_rhsbl_client` *`domain.tld`*
Reject the request when the client hostname is listed with an A record under *`domain.tld`*. See above for additional RBL related configuration parameters.

`check_client_access` *`maptype`*`:`*`mapname`*
`maptype``:`*`mapname`*
Search the named access database for the client hostname, parent domains, client IP address, or networks obtained by stripping least significant octets.

`permit`
`defer`
`reject`
`warn_if_reject`
`reject_unauth_pipelining`
See generic restrictions.

Require HELO (EHLO) command

The `smtpd_helo_required` parameter determines if clients must send a **HELO** (or **EHLO**) command at the beginning of an SMTP session.
Requiring this will stop some UCE software.

Default:

```
smtpd_helo_required = no
```

By default, the Postfix SMTP server does not require the use of **HELO** (**EHLO**).

Syntax:
Specify **yes** or **no**.
Example:

```
smtpd_helo_required = yes
```

HELO (EHLO) hostname restrictions

The `smtpd_helo_restrictions` parameter restricts what hostnames clients may send with the **`HELO`**(**`EHLO`**) command. Some UCE software can be stopped by being strict here.

By default, this restriction is applied when the client sends the RCPT TO command. In order to have the restriction take effect as soon as possible, specify `smtpd_delay_reject = no` in the Postfix **`main.cf`** configuration file. Doing so may cause unexpected results with poorly implemented client software.

Default:

```
smtpd_helo_restrictions =
```

By default, the Postfix SMTP server accepts any garbage in the **HELO** (**EHLO**) command. There is a lot of broken or misconfigured software on the Internet.

Syntax:

Specify a list of zero or more restrictions, separated by whitespace or commas. Restrictions are applied in the order as specified; the first restriction that matches wins.

In addition to restrictions that are specific to HELO (EHLO) command parameters, you may list here any restrictions on the client hostname , client address , sender address or recipient address. The sender or recipient restrictions take effect only if `smtpd_delay_reject = yes` so that all restrictions are evaluated after the RCPT TO command.

Example:

```
smtpd_helo_restrictions = permit_mynetworks, reject_invalid_hostname
```

Restrictions:

`reject_invalid_hostname`
Reject the request when the client HELO or EHLO parameter has a bad hostname syntax. The **`invalid_hostname_reject_code`** specifies the response code to rejected requests (default: 501).

`reject_unknown_hostname`
Reject the request when the hostname in the client HELO (EHLO) command has no DNS A or MX record. The `unknown_hostname_reject_code` specifies the response code to rejected requests (default: **450**).

`reject_non_fqdn_hostname`
Reject the request when the hostname in the client HELO (EHLO) command is not in fully-qualified domain form, as required by the RFC. The `non_fqdn_reject_code` specifies the response code to rejected requests (default: **504**).

`check_helo_access` *`maptype:mapname`*
`maptype:mapname`
Search the named access databasefor the **HELO** hostname or parent domains.

`permit`
`defer`
`reject`
`warn_if_reject`
`reject_unauth_pipelining`
See generic restrictions.

Require strict RFC 821-style envelope addresses
The `strict_rfc821_envelopes` parameter controls how tolerant Postfix is with respect to addresses given in MAIL FROM or RCPT TO commands. Unfortunately, the widely-used Sendmail program tolerates lots of non-standard behavior, so a lot of software expects to get away with it. Being strict to the RFC not only stops unwanted mail, it also blocks legitimate mail from poorly-written mail applications.

Default:
`strict_rfc821_envelopes = no`

By default, the Postfix SMTP server accepts any address form that it can make sense of, including address forms that contain RFC 822-style comments, or addresses not enclosed in <>. There is a lot of broken or misconfigured software out there on the Internet.

Example:
`strict_rfc821_envelopes = yes`

Sender address restrictions

The `smtpd_sender_restrictions` parameter restricts what sender addresses this system accepts in MAIL FROM commands.
By default, this restriction is applied when the client sends the RCPT TO command. In order to have the restriction take effect as soon as possible, specify `smtpd_delay_reject = no` in the Postfix **`main.cf`** configuration file. Doing so may cause unexpected results with poorly implemented client software.

Default:
`smtpd_sender_restrictions =`
By default, the Postfix SMTP server accepts any sender address.

Syntax:
Specify a list of zero or more restrictions, separated by whitespace or commas. Restrictions are applied in the order as specified; the first restriction that matches wins. In addition to restrictions that are specific to sender mail addresses, you can also specify restrictions based on the information passed with the HELO/EHLO command , on the client hostname or network address , or on the recipient address . The recipient restrictions take effect only if `smtpd_delay_reject = yes` so that all restrictions are evaluated after the RCPT TO command.

Example:
```
smtpd_sender_restrictions = hash:/etc/postfix/access,
                            reject_unknown_sender_domain
```

Restrictions:
`reject_unknown_sender_domain`
Reject the request when the sender mail address has no DNS A or MX record.

The unknown_address_reject_code parameter specifies the response code for rejected requests (default: **450**). The response is always **450** in case of a temporary DNS error.

reject_rhsbl_sender *domain.tld*
: Reject the request when the sender mail address domain is listed with an A record under *domain.tld*. The maps_rbl_reject_code parameter specifies the response code for rejected requests (default: **554**), the default_rbl_reply parameter specifies the default server reply, and the rbl_reply_maps parameter specifies tables with server replies indexed by RBL domain.

check_sender_access *maptype*:*mapname*
maptype:*mapname*
: Search the named access database for the sender mail address, sender domain and parent domain, or *localpart*@.

reject_non_fqdn_sender
: Reject the request when the address in the client MAIL FROM command is not in fully-qualified domain form. The non_fqdn_reject_code specifies the response code to rejected requests (default: **504**).

reject_sender_login_mismatch
: Reject the request when $smtpd_sender_owner_maps specifies an owner for the MAIL FROM address, but the client is not (SASL) logged in as that MAIL FROM address owner; or when the client is (SASL) logged in, but the client login name doesn't own the MAIL FROM address according to $smtpd_sender_login_maps.

permit
defer
reject
warn_if_reject
reject_unauth_pipelining
: See generic restrictions.

Recipient address restrictions

The `smtpd_recipient_restrictions` parameter restricts what recipient addresses this system accepts in RCPT TO commands.

Default:

```
smtpd_recipient_restrictions = permit_mynetworks,
                                    reject_unauth_destination
```

By default, the Postfix SMTP server relays mail:

- from trusted clients whose IP address matches `$mynetworks` to any destination,
- from untrusted clients to destinations that match `$relay_domains` or a subdomain thereof, except for addresses that contain sender-specified routing (*`user@elsewhere@domain`*).

In addition to the above, the Postfix SMTP server by default accepts mail for which Postfix is the final destination:

- to destinations that match `$inet_interfaces`,
- to destinations that match `$mydestination`,
- to destinations that match `$virtual_alias_domains`,
- to destinations that match `$virtual_mailbox_domains`.

Syntax:

Specify a list of zero or more restrictions, separated by whitespace or commas. Restrictions are applied in the order as specified; the first restriction that matches wins.

In addition to restrictions that are specific to recipient mail addresses, you can also specify restrictions based on the sender mail address, on the information passed with the HELO/EHLO command , and on the client hostname or network address .

Example:

```
smtpd_recipient_restrictions = permit_mynetworks,
                                    reject_unauth_destination
```

Note: you must specify at least one of the following restrictions: `reject`, `defer`, `defer_if_permit`, *or* `reject_unauth_destination`. *Postfix will refuse to receive mail otherwise.*

Restrictions:

`permit_auth_destination`

Permit the request when one of the following is true:

- the resolved destination address matches `$relay_domains` or a subdomain thereof, and the address contains no sender-specified routing (*`user@elsewhere@domain`*),
- Postfix is the final destination: any destination that matches `$mydestination`, `$inet_interfaces`, `$virtual_alias_domains`, or `$virtual_mailbox_domains`.

`reject_unauth_destination`
Reject the request unless one of the following is true:
- the resolved destination address matches `$relay_domains` or a subdomain thereof, and the address contains no sender-specified routing (*`user@elsewhere@domain`*),
- Postfix is the final destination: any destination that matches `$mydestination`, `$inet_interfaces`, `$virtual_alias_domains`, or `$virtual_mailbox_domains`.

The `relay_domains_reject_code` parameter specifies the response code for rejected requests (default: **554**).

`permit_mx_backup`
Permit the request when the local mail system is MX host for the resolved destination. This includes the case that the local mail system is the final destination. However, the SMTP server will not forward mail with addresses that have sender-specified routing information
(example: *`user@elsewhere@domain`*),

Use the optional `permit_mx_backup_networks` parameter to also require that the primary MX hosts match a list of network blocks.

Relevant configuration parameters: `permit_mx_backup_networks`, `$mydestination`, `$inet_interfaces`.

`check_recipient_access` *`maptype:mapname`*
`maptype:mapname`
Search the named access database for the resolved destination address, recipient domain or parent domain, or *`localpart@`*.

`check_recipient_maps`
Reject the request when the recipient address is not listed in one of the following lookup tables:

Recipient domain matches	Recipient lookup table
`$mydestination` or `$inet_interfaces`	`$local_recipient_maps`
`$virtual_alias_domains`	`$virtual_alias_maps`
`$virtual_mailbox_domains`	`$virtual_mailbox_maps`
`$relay_domains`	`$relay_recipient_maps`

Note 1: a null $local_recipient_maps or $relay_recipient_maps setting means that no recipient check is done for the corresponding domains.

Note 2: Postfix applies an implicit `check_recipient_maps` restriction at the end of all recipient restrictions.

`reject_unknown_recipient_domain`
: Reject the request when the recipient mail address has no DNS A or MX record. The `unknown_address_reject_code` parameter specifies the response code for rejected requests (default: **450**). The response is always **450** in case of a temporary DNS error.

`reject_rhsbl_recipient` *`domain.tld`*
: Reject the request when the recipient mail address domain is listed with an A record under *`domain.tld`*. The `maps_rbl_reject_code` parameter specifies the response code for rejected requests (default: **554**), the `default_rbl_reply` parameter specifies the default server reply, and the `rbl_reply_maps` parameter specifies tables with server replies indexed by RBL domain.

`reject_non_fqdn_recipient`
: Reject the request when the address in the client RCPT TO command is not in fully-qualified domain form. The `non_fqdn_reject_code` specifies the response code to rejected requests (default: **504**).

`permit`
`defer`
`reject`
`warn_if_reject`
`reject_unauth_pipelining`
: See generic restrictions.

ETRN command restrictions

Not really an UCE restriction, the `smtpd_etrn_restrictions` parameter restricts what domains can be specified in ETRN commands, and what clients can issue ETRN commands.

Default:

```
smtpd_etrn_restrictions =
```

By default, the Postfix SMTP server accepts any ETRN command from any client.

Syntax:

Specify a list of zero or more restrictions, separated by whitespace or commas. Restrictions are applied in the order as specified; the first restriction that matches wins.

In addition to restrictions that are specific to ETRN domain names, you can also specify restrictions based on the information passed with the HELO/EHLO command , and on the client hostname or network address .

Example:

```
smtpd_etrn_restrictions = permit_mynetworks,
                          hash:/etc/postfix/etrn_access, reject
```

Restrictions:

`check_etrn_access` *`maptype:mapname`*
`maptype:mapname`

Search the named access databasefor the domain specified in the **ETRN** command, or its parent domains. Reject the request if the result is **REJECT** *text...*or "`[`**`45`**`]`*`XX text`*". Permit the request if the result is **OK**or **RELAY**or all-numerical. Otherwise, treat the result as another list of UCE restrictions. The `access_map_reject_code` parameter specifies the result code for rejected requests (default: **554**).

`permit`
`defer`
`reject`
`warn_if_reject`
`reject_unauth_pipelining`

See generic restrictions.

Generic restrictions

The following restrictions can use used for client hostnames or addresses, for HELO (EHLO) hostnames, for sender mail addresses and for recipient mail addresses.

Restrictions:

`permit`
: Permit the request. This restriction is useful at the end of a restriction list, to make the default policy explicit.

`defer`
: Defer the request. The client is told to try again later. This restriction is useful at the end of a restriction list, to make the default policy explicit.

`reject`
: Reject the request. This restriction is useful at the end of a restriction list, to make the default policy explicit. The `reject_code` configuration parameter specifies the response code to rejected requests (default: **554**).

`warn_if_reject`
: Change the meaning of the next restriction, so that it logs a warning instead of rejecting a request (look for logfile records that contain "`reject_warning`"). This is useful for testing new restrictions in a "live" environment without risking unnecessary loss of mail.

`reject_unauth_pipelining`
: Reject the request when the client sends SMTP commands ahead of time without knowing that Postfix actually supports SMTP command pipelining. This stops mail from bulk mail software that improperly uses SMTP command pipelining to speed up deliveries.

Additional UCE control parameters

`default_rbl_reply`
: The default reply template that is used when an SMTP client request is blocked by a `reject_rbl` or `reject_rhsbl` restriction. The reply template is subjected to exactly one level of $name macro substitution as described below. The `smtpd_expansion_filter` configuration parameter specifies the set of characters that are allowed in $name macro expansions. Characters outside the allowed set are replaced by "_".

  Default:

  ```
  default_rbl_reply = $rbl_code Service unavailable;
  $rbl_class [$rbl_what] blocked using $rbl_domain$
  {rbl_reason?; $rbl_reason}
  ```

  Instead of the form $*name* you can also specify ${*name*} or $(*name*).

Macro expansion syntax:

`$client`
: The client hostname and IP address, formatted as *name*[*address*].

`$client_name`
: The client hostname, or **unknown**.

`$client_address`
: The client IP address.

`$helo_name`
: The hostname given in the HELO or EHLO command, or the empty string when no HELO or EHLO command was given.

`$sender`
: The sender address, or <> in case of the null address.

`$sender_name`
: The sender address localpart, or <> in case of the null address.

`$sender_domain`
: The sender address domain, or the empty string when no domain is available.

`$recipient`
: The recipient address, or <> in case of the null address.

`$recipient_name`
: The recipient address localpart, or <> in case of the null address.

`$recipient_domain`
: The recipient address domain, or the empty string when no domain is available.

`$rbl_what`
: The blacklisted entity: an IP address, a hostname, a domain name, or an email address whose domain is blacklisted.

`$rbl_domain`
: The RBL domain where $rbl_what is blacklisted with an A record.

`$rbl_reason`
: The reason why $rbl_what is blacklisted, or the empty string when no information is available.

`$rbl_class`
: The blacklisted entity type: Client host, Helo command, Sender address, or Recipient address.

`$rbl_code`
: The numerical server reply code, as specified with the `maps_rbl_reject_code` configuration parameter (default: 554).

`All other text`
: Copied without change, with the exception of conditional macro expansion as described below.

Conditional macro expansion syntax:

`${`*`name`*`?`*`text`*`}`
: expands to *text* if $*name* is not empty.

`${`*`name`*`:`*`text`*`}`
: expands to *text* if $*name* is empty.

permit_mx_backup_networks

Restrict the use of the permit_mx_backup relay control feature to destinations whose primary MX hosts match a list of network blocks.

Default:

```
permit_mx_backup_networks =
```

That is, all networks are authorized by default.

Syntax:

Specify a list of network blocks in CIDR (network/mask) notation, for example:

```
permit_mx_backup_networks = 168.100.0.0/16
```

You can also specify the absolute pathname of a pattern file instead of listing the patterns in the **main.cf** file.

rbl_reply_maps

This parameter specifies lookup tables with RBL reply templates indexed by RBL domain name. If no template is found, the default_rbl_reply template is used instead.

Default:

rbl_reply_maps =

By default, Postfix always uses the default_rbl_reply template.

Syntax:

Specify zero or more *type*:*name* lookup tables, separated by whitespace and/or commas. For the syntax of the template reply strings, see the default_rbl_reply parameter description.

relay_domains

This parameter controls the behavior of the reject_unauth_destination and permit_auth_destination restrictions that can appear as part of a recipient address restriction list.

Default:

```
relay_domains = $mydestination
```

By default, the Postfix SMTP server relays mail:

- from trusted clients whose IP address matches $mynetworks,
- from untrusted clients to destinations that match $relay_domains or a subdomain thereof, except for addresses that contain sender-specified routing (*user@elsewhere@domain*).

Syntax:

Specify zero or more domain names, */file/name* patterns and/or *type*:*name* lookup tables, separated by whitespace and/or commas.
A */file/name* is replaced by its contents; *type*:*name* requests that table lookup is done instead of string comparison.
A host or destination address matches $relay_domains when its name or parent domain matches any of the names, files or lookup tables listed in $relay_domains.

smtpd_sender_login_maps

This parameter specifies ownership of MAIL FROM addresses, as used by the `reject_sender_login_mismatch` sender address restriction.

Default:

`smtpd_sender_login_maps =`

Syntax:

Specify zero or more type:name lookup tables, separated by whitespace and/or commas. The maps are searched in the specified order. Regexp tables are allowed.

Each map entry specifies a sender address and the login name that owns the address. The search order is:

`user@domain owner`

This form has the highest precedence.

`user owner`

This matches *user@site* when *site* is equal to `$myorigin`, when *site* is listed in `$mydestination`, or when it is listed in `$inet_interfaces`.

`@domain owner`

This matches every address in the specified domain, and has the lowest precedence.

Postfix Configuration - Address Manipulation

Introduction

Although the initial Postfix release has no address rewriting language, it can do quite a bit of address manipulation via table lookup. While a message flows through the Postfix system, its addresses are mangled in the order described in this document.

Unless indicated otherwise, all parameters described here are in the **main.cf** file. If you change parameters of a running Postfix system, don't forget to issue a **postfix reload** command.

All mail:

- Rewrite addresses to standard form
- Canonical address mapping
- Address masquerading
- Virtual address mapping
- Mail transport switch
- Relocated users table

Local delivery:

- Alias database
- Per-user .forward files
- Non-existent users

Rewrite addresses to standard form

Before the `cleanup` daemon runs an address through any lookup table, it first rewrites the address to the standard *`user@fully.qualified.domain`* form, by sending the address to the `trivial-rewrite` daemon. The purpose of rewriting to standard form is to reduce the number of entries needed in lookup tables. The Postfix `trivial-rewrite` program implements the following hard-coded address manipulations:

Rewrite *`@hosta,@hostb:user@site`* to *`user@site`*
: The source route feature has been deprecated. Postfix has no ability to handle such addresses, other than to strip off the source route.

Rewrite *`site!user`* to *`user@site`*
: This feature is controlled by the boolean **`swap_bangpath`** parameter (default: **`yes`**). The purpose is to rewrite **UUCP**-style addresses to domain style. This is useful only when you receive mail via **UUCP**, but it probably does not hurt otherwise.

Rewrite *`user%domain`* to *`user@domain`*
: This feature is controlled by the boolean **`allow_percent_hack`** parameter (default: **`yes`**). Typically, this is used in order to deal with monstrosities such as *`user%domain@otherdomain`*.

Rewrite *user* to *user@$myorigin*
: This feature is controlled by the boolean append_at_myorigin parameter (default: **yes**). The purpose is to get consistent treatment of *user* on every machine in $myorigin.

  You probably should never turn off this feature, because a lot of Postfix components expect that all addresses have the form *user@domain*.

  If your machine is not the main machine for **$myorigin** and you wish to have some users delivered locally without going via that main machine, make an entry in the virtual table that redirects *user@$myorigin* to *user@$myhostname*.

Rewrite *user@host* to *user@host.$mydomain*
: This feature is controlled by the boolean append_dot_mydomain parameter (default: **yes**). The purpose is to get consistent treatment of different forms of the same hostname.

  Some will argue that rewriting *host* to *host.$mydomain* is bad. That is why it can be turned off. Others like the convenience of having the local domain appended automatically.

Rewrite *user@site.* to *user@site* (without the trailing dot).

Canonical address mapping

Before the cleanup daemon stores inbound mail into the **incoming** queue, it uses the canonical table to rewrite all addresses in message envelopes and in message headers, local or remote. The mapping is useful to replace login names by *Firstname.Lastname* style addresses, or to clean up invalid domains in mail addresses produced by legacy mail systems.

Canonical mapping is disabled by default. To enable, edit the canonical_maps parameter in the **main.cf** file and specify one or more lookup tables, separated by whitespace or commas. For example:

```
canonical_maps = hash:/etc/postfix/canonical
```

In addition to the canonical maps which are applied to both sender and recipient addresses, you can specify canonical maps that are applied only to sender addresses or to recipient addresses. For example:

```
sender_canonical_maps = hash:/etc/postfix/sender_canonical

recipient_canonical_maps = hash:/etc/postfix/recipient_canonical
```

The sender and recipient canonical maps are applied before the common canonical maps.

Sender-specific rewriting is useful when you want to rewrite ugly sender addresses to pretty ones, and still want to be able to send mail to the those ugly address without creating a mailer loop.

Address masquerading

Address masquerading is a method to hide all hosts inside a domain behind their mail gateway, and to make it appear as if the mail comes from the gateway itself, instead of from individual machines.

Address masquerading is disabled by default. To enable, edit the **`masquerade_domains`** parameter in the **`main.cf`** file and specify one or more domain names separated by whitespace or commas. The list is processed left to right, and processing stops at the first match. Thus,

```
masquerade_domains = foo.example.com example.com
```

strips `any.thing.foo.example.com` to `foo.example.com`, but strips `any.thing.else.example.com` to `example.com`.

A domain name prefixed with **`!`** means do not masquerade this domain or its subdomains. Thus,

```
masquerade_domains = !foo.example.com example.com
```

does not change any.thing.foo.example.com and foo.example.com, but strips any.thing.else.example.com to example.com.

The `masquerade_exceptions` configuration parameter specifies what user names should not be subjected to address masquerading. Specify one or more user names separated by whitespace or commas. For example,

```
masquerade_exceptions = root
```

By default, Postfix makes no exceptions.

Subtle point: by default, address masquerading is applied only to message headers and to envelope sender addresses, but not to envelope recipients. This allows you to use address masquerading on a mail gateway machine, while still being able to forward mail from outside to users on individual machines.

In order to subject envelope recipient addresses to masquerading, too, specify (only available with Postfix versions after 20010802):

```
masquerade_classes = envelope_sender, envelope_recipient,
                     header_sender, header_recipient
```

If you do this, Postfix will no longer be able to send mail to individual machines.

Virtual address aliasing

After applying the canonical and masquerade mappings, the `cleanup` daemon uses the `virtual alias` table to redirect mail for all recipients, local or remote. The mapping affects only envelope recipients; it has no effect on message headers or envelope senders. Virtual alias lookups are useful to redirect mail for simulated virtual domains to real user mailboxes, and to redirect mail for domains that no longer exist. Virtual alias lookups can also be used to transform *`Firstname.Lastname`* back into UNIX login names, although it seems that local `aliases` are a more appropriate vehicle.

Virtual aliasing is disabled by default. To enable, edit the `virtual_alias_maps` parameter in the **`main.cf`** file and specify one or more lookup tables, separated by whitespace or commas. For example:

```
virtual_alias_maps = hash:/etc/postfix/virtual
```

Addresses found in virtual alias maps are subjected to another iteration of virtual aliasing, but are not subjected to canonical mapping, in order to avoid loops.

Mail transport switch

Once the address rewriting and resolving daemon has established the destination of a message, it determines the default delivery method for that destination. Postfix distinguishes four major address classes, each with its own default delivery method.

Destination matches	Default delivery agent	Controlling parameter
`$mydestination` or `$inet_interfaces`	local	`$local_transport`
`$virtual_mailbox_domains`	virtual	`$virtual_transport`
`$relay_domains`	relay (clone of smtp)	`$relay_transport`
none	smtp	`$default_transport`

The optional `transport` table overrides the default message delivery method (this table is used by the address rewriting and resolving daemon). The transport table can be used to send mail to specific sites via **UUCP**, or to send mail to a really broken mail system that can handle only one SMTP connection at a time (yes, such systems exist and people used to pay real money for them).

Transport table lookups are disabled by default. To enable, edit the **`transport_maps`** parameter in the **`main.cf`** file and specify one or more lookup tables, separated by whitespace or commas. For example:

```
transport_maps = hash:/etc/postfix/transport
```

Relocated users table

Next, the address rewriting and resolving daemon runs each recipient name through the `relocated` database. This table provides information on how to reach users that no longer have an account, or what to do with mail for entire domains that no longer exist. When mail is sent to an address that is listed in this table, the message is bounced with an informative message.

Lookups of relocated users are disabled by default. To enable, edit the `relocated_maps` parameter in the **`main.cf`** file and specify one or more lookup tables, separated by whitespace or commas. For example:

```
relocated_maps = hash:/etc/postfix/relocated
```

Alias database

When mail is to be delivered locally, the `local` delivery agent runs each local recipient name through the `aliases` database. The mapping does not affect addresses in message headers. Local aliases are typically used to implement distribution lists, or to direct mail for standard aliases such as **`postmaster`** to real people. The table can also be used to map *`Firstname.Lastname`* addresses to login names.

Alias lookups are enabled by default. The default configuration depends on the system environment, but it is typically one of the following:

```
alias_maps = hash:/etc/aliases
alias_maps = dbm:/etc/aliases, nis:mail.aliases
```

The path to the alias database file is controlled via the **`alias_database`** configuration parameter. The value is system dependent. Usually it is one of the following:

```
alias_database = hash:/etc/aliases       (4.4BSD, LINUX)
alias_database = dbm:/etc/aliases        (4.3BSD, SYSV<4)
alias_database = dbm:/etc/mail/aliases   (SYSV4)
```

For security reasons, deliveries to command and file destinations are performed with the rights of the alias database owner. A default userid, `default_privs`, is used for deliveries to commands/files in *root*-owned aliases.

Per-user `.forward` files

Users can control their own mail delivery by specifying destinations in a file called `.forward` in their home directories. The syntax of these files is the same as with system aliases, except that the lookup key and colon are not present.

Non-existent users

When the local delivery agent finds that a message recipient does not exist, the message is normally bounced to the sender ("user unknown"). Sometimes it is desirable to forward mail for non-existing recipients to another machine. For this purpose you can specify an alternative destination with the `luser_relay` configuration parameter.
Alternatively, mail for non-existent recipients can be delegated to an entirely different message transport, as specified with the `fallback_transport` configuration parameter. For details, see the `local` delivery agent.

Note: if you use the `luser_relay` feature in order to receive mail for non-UNIX accounts, then you must specify:

```
local_recipient_maps =
```

(i.e. empty) in the **`main.cf`** file, otherwise the Postfix SMTP server will reject mail for non-UNIX accounts with "User unknown in local recipient table".

`luser_relay` can specify one address. It is subjected to *$name* expansions. The most useful examples are:

`$user@other.host`
: The bare username, without address extension, is prepended to *@other.host*.
For example, mail for *username+foo* is sent to *username@other.host*.

`$mailbox@other.host`
: The entire original recipient localpart, including address extension, is prepended to *@other.host*.
For example, mail for *username+foo* is sent to *username+foo@other.host*.

`sysadmin+$user`
: The bare username, without address extension, is appended to *sysadmin*.
For example, mail for *username+foo* is sent to *sysadmin+username*.

`sysadmin+$mailbox`
: The entire original recipient localpart, including address extension, is appended to *sysadmin*.
For example, mail for *username+foo* is sent to *sysadmin+username+foo*.

Mail Statistics with 'Awstats'

AWStats is a statistics program that generates web sites that displays statistics based on web logs , FTP logs or Mail logs. It is written in perl and can be run as a CGI or as a standalone command(normally as a cron job).

The following instructions are concerning uniquely the running of awstats as a standalone program. Since SuSE doesn't provide AWSTats in their installation CDs/DVD this installation example was made on Debian Sarge.

- **Installing awstats**
  - Run the command
    `apt-get install awstats`
    The help files are located in `/usr/share/doc/awstats/html`
  - Create 2 sub directories of the web site(script output) called `awstats-icon and cgi-bin` eg.
    ```
    mkdir  /home/www/mydomain.com/mailstats/awstats-icon
    mkdir  /home/www/mydomain.com/mailstats/cgi-bin
    ```
  - Copy the main script (`awstats.pl`) to the */websitePath*`/cgi-bin` eg.
    ```
    cp /usr/lib/cgi-bin/awstats.pl \
    /home/www/mydomain.com/mailstats/cgi-bin/
    ```

- **Configuring awstats**
  The main configuration file is:
  `/etc/awstats/awstats.conf`
  it is the default config file. If another config file is prefered then the command line needs to include the following parameter:
  eg. .... `-config=mail`
  which will tell `awstats.pl` to use the configuration file:
  `/etc/awstats/awstats.mail.conf`

- **Setting AWSTats for Mail Statistics**

  You must setup AWStats to use a mail log file preprocessor (*`maillogconvert.pl`* is provided into AWStats *tools* directory, but you can use the one of your choice):
  For this, copy config *"`awstats.model.conf`"* file to *"`awstats.mail.conf`"*.
  Modify this new config file: For standard Postfix, Sendmail, MDaemon and standard QMail logfiles, set

```
LogFile="perl /path/to/maillogconvert.pl standard < /pathtomaillog/maillog |"
```

If the logfiles are compressed, they can be processed this way

```
LogFile="gzip -cd /var/log/maillog.0.gz|/path/to/maillogconvert.pl standard |"
```

Then, whatever is you mail server, you must also change:

```
SiteDomain="mydomain.com"
Lang="de"          (only if German language is desired)

LogType=M
LogFormat="%time2 %email %email_r %host %host_r %method %url %code %bytesd"
LevelForBrowsersDetection=0
LevelForOSDetection=0
LevelForRefererAnalyze=0
LevelForRobotsDetection=0
LevelForWormsDetection=0
LevelForSearchEnginesDetection=0
LevelForFileTypesDetection=0
ShowMenu=1
ShowSummary=HB
ShowMonthStats=HB
ShowDaysOfMonthStats=HB
ShowDaysOfWeekStats=HB
ShowHoursStats=HB
ShowDomainsStats=0
ShowHostsStats=HBL
ShowAuthenticatedUsers=0
ShowRobotsStats=0
ShowEMailSenders=HBML
ShowEMailReceivers=HBML
ShowSessionsStats=0
ShowPagesStats=0
ShowFileTypesStats=0
ShowFileSizesStats=0
ShowBrowsersStats=0
ShowOSStats=0
ShowOriginStats=0
ShowKeyphrasesStats=0
ShowKeywordsStats=0
ShowMiscStats=0
ShowHTTPErrorsStats=0
ShowSMTPErrorsStats=1
```

- **Running AWStats from the command line:**
  The command line needs to following format:

  ```
  perl /path/to/awstats.pl -config=xxxx options
  ```

  eg.
  ```
  perl /home/www/mydomain.com/mailstats/cgi-bin/awstats.pl \
       -config=mail \
       -update \
       -output > /home/www/mydomain.com/mailstats/index.html
  ```

  This command will update (-update) only the new data from the already processed, it will use the configuration file `/etc/awstats/awstats.mail.conf` and will create the report in html format in:
  `/home/www/mydomain.com/mailstats/index.html`

- **Configuring Apache for reading the results**
  Because some of the links placed into this web page are running the cgi `awstats.pl`, Apache needs to be configured accordingly.

  eg.
  ```
  <VirtualHost 153.67.246.28>
       ServerName mailstats.mydomain.com
       DocumentRoot /home/www/mydomain.com/mailstats
       <Directory /home/www/mydomain.com/mailstats>
  ```

```
            DirectoryIndex index.html
            Allow from All
            AuthName "Mail Statistics“
            AuthType Basic
            AuthUserFile /home/www/mywebsite/auth_users
            Require user martin aline
            Satisfy all
        </Directory>
        <Directory  /home/www/mydomain.com/mailstats/cgi-bin>
            AllowOverride None
            options ExecCGI
            SetHandler cgi-script
        </Directory>
    </VirtualHost>
```

-

Using Postfix

A basic guide on configuring and installing the Postfix mail server.

By Alan P. Laudicina

Introduction

Tired of the sendmail's cryptic configuration, or do you find yourself complaining about its speed? Well then, postfix could be the MTA for you. The Postfix website defines postfix as a MTA which "attempts to provide an alternative to the widely-used Sendmail program." If it's speed and security you're looking for, Postfix is a very nominal choice for a MTA. According to the project's web site, Postfix is up to three times faster than its closest competitor, boasting the capability to send up to 1,000,000 *different* messages in a day. The MTA uses multiple layers of defense to protect the local system against intruders, as well as having the ability to run in a chroot jail. Installing on most operation systems is a trivial procedure, although in FreeBSD installation should be done differently to avoid the overwriting of the binaries when a `make world` is done. Another way to avoid this is to use a mail wrapper. (For more information on mail wrappers read the "Mail Wrappers" heading under the Installation section.)

Configuration

All of the many configuration parameters can be found in the main.cf file, located in the `./conf` directory in the postfix source. You need not change every parameter, as they are set to sensible defaults. Here are the details on some of the more important parameters, which will affect the performance of Postfix the most. Please note that if you change the main.cf file after installation, you must issue the `postfix reload` command. After installation, the **`main.cf`** file can be found in the /etc/postfix directory.

- `queue_directory` - the location of the Postfix queue as well as the root dir of the postfix daemons that run chrooted. This field should be left with the default `/var/spool/postfix`

- `daemon_directory` - the location of the daemon programs such as smptd, pickup, cleanup, etc.

- `mail_owner` - the owner of Postfix's queue and most of the daemon processes. For this you must add a user to your machine, this has to be a user that owns no other files or processes (so using nobody here is a very bad idea for security reasons).

- `myorigin` - the origin is set to `$myhostname` by default, which defaults to the local hostname of the machine. This should not be used unless you are running a very small site. Most people want to change `myorigin` to `$mydomain` which will default to the parent domain of the machine name
(i.e. if the hostname is `lame.unixpower.org` and you are using `$myhostname`, the origin will be `lame.unixpower.org`. On the other hand if you were using `$mydomain`, the origin will be `unixpower.org`.)

- `inet_interfaces` - the inet_interfaces parameter defines which network interface addresses that the stmp daemon will listen on. By default this is set to all, which will listen on any active interface on the machine. This will control the delivery to `users@<IP>`.

- `mydestination` - this parameter specifies the list of domains that the machine considers *itself*. The default of `$myhostname` and `localhost.$mydomain` should do here. Don't specify the virtual domains that are hosted on the machine here!

- `mailbox_command` **-** this parameter defines the external command to use instead of local mailbox delivery. It is a completely optional parameter. If you're interested in having `procmail` to do your mail, this is where you set it.

- `mynetworks` - mynetworks specifies a certain list of network addresses that are local to this machine. The list is used to distinguish users from strangers. The addresses go in the format of `X.X.X.0/X` and can be separated by a comma. By default the list of all of the networks attached to the machine is a complete class A network (`X.0.0.0/8`), a complete class B network (`X.X.0.0/16`), a complete class C network (`X.X.X.0/24`), and so on. You can also specify a path of a pattern file instead of listing the patterns here.

Compilation

The compilation of Postfix is a very fast and easy task. In BSD, the only thing you will need to do is go to the main postfix directory and type `make`. Compiling Postfix is much faster on my machine then compiling sendmail, taking only a minute and fifty seconds (on a Pentium II 300 with 160mb of RAM). Sendmail takes approximately a minute more than compiling Postfix on the same machine.

Installation

After the configuration and compilation of Postfix, installation is the last step. To install Postfix on a BSD machine, you must first move the sendmail binaries so that you can replace the files without overwriting them. To do this you can su to root and execute the following commands:

```
# mv /usr/sbin/sendmail /usr/sbin/sendmail.old
# mv /usr/bin/mailq /usr/bin/mailq.old
# mv /usr/bin/newaliases /usr/bin/newaliases.old
# chmod 755 /usr/sbin/sendmail.old /usr/bin/mailq.old /usr/bin/newaliases.old
```

Note: After a `make world` to your BSD system, the Postfix binaries will be replaced with sendmail libraries. This makes it a **very** good idea to not delete the Postfix source tree after compilation, so in the future after a `make world` you can always come back and repeat the steps for the installation of the Postfix binaries listed above.

Mail Wrappers

Some BSD machines may pack with a mail wrapper. It is used so that you can easily have several MTAs installed at the same time. The mail wrapper is not required, but if you plan to use it, you should definitely read the `mailwrapper`(8) and `mailer.conf`(5) man pages. Instead of replacing the sendmail binaries, you could simply setup the `/etc/mailer.conf` (or `/etc/mail/mailer.conf`) with something like:

```
# Emulate sendmail using postfix
sendmail        /usr/libexec/postfix/sendmail
send-mail       /usr/libexec/postfix/sendmail
mailq           /usr/libexec/postfix/sendmail
newaliases      /usr/libexec/postfix/sendmail
```

After the installation of the Postfix binaries you must create the user that postfix will run as. This user is to be named 'postfix' and have a unique user and group id, with a non-existent shell (so that nobody can login to the account for security reasons), the account does not require to have an existing home directory either. To add the account to my machine, I executed the following commands:

```
# echo "postfix:*:33333:33333:Postfix Mail Daemon:/nonexistant:/sbin/nologin" \
           >> /etc/passwd
# echo "maildrop:*:33335:" >> /etc/group
```

(Before you add the 'postfix' user and the 'maildrop' group, you may want to make sure the uid and gid I use are available. To do this look through the /etc/passwd and /etc/group files with a command like more /etc/passwd or more /etc/group. You may also use the useradd(8) command.)

After you add the user that the mail daemon will run as, it is a good idea to forward all that user's email to root. We do this because nobody can login as the user postfix, so it is a good idea to forward any email it gets to root. Here is how you add the alias:

```
# echo "postfix: root" >> /etc/aliases
```

Now comes a decision for the person who is installing postfix from the directions I am giving. If a world-writable maildrop is okay with you, you can skip the next section and go to the "sh INSTALL.sh" section. If you want to protect the maildrop directory, read the following section.

Protecting your Maildrop directory

By default, postfix installs with a world-writable, mode 1733, sticky maildrop so that local users can submit mail. Well this method avoids using set-[gu]id software, it is usually a bad idea if you have some annoying lusers. The world-writable maildrop would allow those users to fill the maildrop directory with masses of garbage and possibly crash the mail system. So to avoid this, we will add another group that is unique suck as the 'postfix' group. You can do this with the following command:

```
# echo "maildrop:*:33335:" >> /etc/group
```

After you add the maildrop group, you can proceed to the next section.

sh INSTALL.sh

If you have made it this far, you are ready to start the "real" installation program. You can do this by going to the top level directory of the postfix source and executing the following command:

```
# sh INSTALL.sh
```

This will run you through a script that will ask for input. The defaults are fine here until you get the the "`setgid: [no]`" option. When you get here if you followed section 5, then you want to replace the no by typing "`maildrop`" and then pressing enter. If you skipped section 5 and are installing with a non-protected maildrop directory, then you can just leave this with the default "no" option. After this step the "`manpages`" option should also be left with the default selection.

Replacing sendmail forever

This document teaches how to replace sendmail forever on the BSD system. To do this we are going to need to kill the sendmail daemon and restart it so that it only sends out the messages it may have queued. To do this you want to execute the following commands:

```
# kill -9 `ps ax | grep '[s]endmail' | awk '{ print $1 }'`
# /usr/sbin/sendmail.old -q
# postfix start
```

Postfix can be started using the same syntax as sendmail, so it is not required to change the /etc/rc.conf file. When first run you should watch the syslog for complaints from Postfix. Since we changed the main.cf file previously, you should now have a completely running mail daemon. You can find all the configuration files in /etc/postfix. When you modify any of these files you must reload the daemon using `postfix reload` as root.

Using White Listing

I'm using one of the blacklists to block spam and it's working fine. Now one of our customers/partners has got themselves listed, so my mail server is dutifully rejecting their messages. Is there a way to allow just their messages but still use the blacklist?

You can create a whitelist that will accept messages from certain addresses or domains. For example:

```
#
# main.cf
#
smtpd_recipient_restrictions =
        permit_mynetworks
        reject_unauth_destination
        ...
        check_sender_access hash:/etc/postfix/whitelist
        reject_rbl_client dnsbl.njabl.org
        ...

#
# whitelist
#
@customer_domain.com    OK
```

Make sure the whitelist check occurs before the `reject_rbl_client` check. Remember that email addresses are easily faked. Whenever you add whitelisting to your configuration be very careful that you don't expose your server to open relaying. Make sure that your whitelisting occurs after `reject_unauth_destination` (or another rejection restriction).

MAILDIR Mailbox configuration:
Normally the mailbox is in `/var/mail/username` in 'mbox' format.
To change the mailbox type to **Maildir** Format do the following:
- In `/etc/postfix/main.cf`:
  Make sure the directive 'mailbox_command' is as follows:
  `mailbox_command = procmail -a "$EXTENSION"`
- Add the `~/.procmailrc` file with the following content(NOT `/etc/procmailrc`):

```
MAILDIR=$HOME/Maildir
:0
$MAILDIR/
```

- Add a copy of the file `~/.procmailrc` `/etc/skel/.procmailrc`
  Add the additional directory: `/etc/skel/Maildir/`
  and the following subdirectories: `/etc/skel/Maildir/cur`
  `/etc/skel/Maildir/new`
  `/etc/skel/Maildir/tmp`
- Create the same structure for each existing user. eg.
  `/home/username/Maildir/`
  `/home/username/Maildir/cur`
  `/home/username/Maildir/new`
  `/home/username/Maildir/tmp`
  and give their ownership to the user.
  `chown -R username. /home/username/Maildir/`
- Add a copy of the file `~/.procmailrc` `/home/username/.procmailrc`

- If the `dovecot-imapd` is used, Make sure it is configured accordingly:

```
/etc/dovecot/dovecot.conf
      protocols = imap
      mail_location = maildir:~/Maildir
      maildir_copy_with_hardlinks=yes
```

- No special changes needed for `squirrelmail`

Problems with Debian Amavis and ClamAV Daemon

UPDATE: Since I wrote this HOWTO, I found there is a very simple way to fix the file permission issues without performing all the user changes and file ownership changes I have listed below in the original HOWTO. The original HOWTO may however still provide insight into other clamd.conf and freshclam.conf configuration options.

One requirement for a successful installation is **'AllowSupplementaryGroups yes' must be included in clamd.conf**. Another requirement is **the value after CONTSCAN in amavisd.conf must match the LocalSocket parameter in clamd.conf** (change amavisd.conf if it does not). A third requirement is **TCPSocket cannot be used simultaneously with LocalSocket** so TCPSocket must be commented out and LocalSocket must be enabled. The group that your amavisd-new user belongs to must also have write privileges to the amavisd-new user's home directory and subdirectories. This step should have been done during the installation of amavisd-new, and would consist of doing something similar to **chmod -R 750 /var/amavis** or **chmod -R 750 /var/lib/amavis** (adjust path as needed). Once you have ClamAV installed and the clamav user and clamav group have been created and the above requirements have been met, all you may need to do is make the user "clamav" a member of the same group that the amavisd-new user belongs to. Your amavisd-new user likely belongs to the "amavis" or "vscan" group. If that is the case you would issue the command:

```
gpasswd -a clamav amavis
(or)
gpasswd -a clamav vscan (for example)
```

You can test that clamav now belongs to both groups by issuing the command "groups clamav". The command above may not bring the desired result on some systems, so as an alternative you can directly edit `/etc/group` (use vigr if it's installed and you are familiar with vi commands) and manually add the user "clamav" to the "amavis" or "vscan" group:

```
amavis:x:104:clamav
(or)
vscan:x:999:clamav (for example)
```

As a third alternate, you could (for example) possibly use `usermod -G amavis clamav` but if you do, be very careful that you use an upper case "G" or you will have a mess to fix. Then, of course, stop and restart clamd and amavisd (amavisd-new), or simply reboot (if appropriate). Send a test virus through and read the log files. I suggest downloading eicar.com.txt, renaming it to eicar.txt and then attaching it to the email. Give it a try. If it doesn't work, try the other "change owner and ownership" method outlined in the original HOWTO below. Also consider that SELinux or AppArmor may interfere with the way clamd and amavisd-new work together. If you use SELinux or AppArmor I leave it up to you to solve that problem. This document assumes the reader knows to comment out "@bypass_virus_checks_*" to enable virus scanning (and to also uncomment the "ClamAV-clamd" code in the @av_scanners section). One last note: in at least one version of the 0.90 release, it can take several minutes for clamd to create the Unix socket. If you are using a 0.90 version, please allow several minutes for creation of the clamd socket once clamd is started. Better yet, upgrade to the latest version.

And now the original HOWTO:

It seems many people get frustrated when trying to configure ClamAV to work with amavisd-new. They get the ClamAV daemon (clamd) installed via their distro's package maintainer or they download the source and install it from there. Part of the frustration comes from the inconsistent placement of files between different

versions of ClamAV and different versions of binary packages available, but this can be said of nearly any program that consists of more than a few files. Partly because of these inconsistencies it becomes difficult for anyone to instruct a person on how to configure ClamAV for use with amavisd-new.

If you have the opportunity, you should install the binary package available for your distribution. Binary packages are available for Debian, RedHat Fedora, PLD Linux Distribution, Mandrake, Slackware, FreeBSD, OpenBSD, NetBSD and AIX. Installing and configuring ClamAV from source code is somewhat more daunting and you will have to come up with way to start clamd automatically and automate the virus definition database updates. I suggest you read through this document, then read the ClamAV documentation.

I suggest running `updatedb` and then `locate clam | more` and `locate .cvd` to find where the files are located. If you would like to move some of the data files that ClamAV uses (the ones that are referred to in the configuration files) you can create new directories and move the files there provided you also make the changes in the configuration files and change the ownership of the new directories (and the files contained therein).

Almost all the problems with clamd (as it relates to amavisd-new) stem from file permission issues or an incorrectly configured LocalSocket. From what I see, when clamd is installed, the "clamav" user that is created (either manually or by the installation process) is the only "normal" user that can write to the files that the program uses during it's operation. Thus, when you install the clamd daemon the first time, and you try to use it with amavisd-new, you may get "Can't connect to UNIX socket". This is because you are running amavisd-new as a different user (probably "amavis" or "vscan" or something) and that user does not have permission to write to a file that the two programs use to communicate with each other (the LocalSocket file).

I imagine you could break all the security that ClamAV has set up and allow anyone to write it's files, but I don't want to break stuff. One alternative is to set ClamAV up to run under the same user that amavisd-new runs under and then hand the ownership of the ClamAV files over to that user. Let's call that user "amavis" from now on. Fortunately, the ClamAV developers expected there might be instances where doing this might be necessary so they built the capability into the program. **So our somewhat simple task is to change the owner the program runs under, then change the ownership of the files that it writes to.**

The examples below are from a Debian machine on which I installed clamav-daemon version 0.90.1-1 using "apt-get -t unstable install clamav clamav-daemon". Use the following directory names and file names and user names **only as examples**. They are provided to illustrate the concepts and your system may use different directories, file names and user names.

Open up your `/etc/clamav/clamd.conf` with your favorite editor.
This is the clamav main configuration file. Look for a line similar to this:
`LocalSocket` **`/var/run/clamav/clamd.ctl`**
Make a note of this.

Now open up your amavisd.conf, mine is `/etc/amavis/amavisd.conf`
and look for the section:

```
 ['Clam Antivirus-clamd',
  \&ask_daemon, ["CONTSCAN {}\n", "/var/run/clamav/clamd.ctl"],
  qr/\bOK$/, qr/\bFOUND$/,
  qr/^.*?: (?!Infected Archive)(.*) FOUND$/ ],
```

The text illustrated above must match the LocalSocket parameter you found in clamd.conf.

Edit amavisd.conf to match what you found in clamd.conf if it is different.
This "clamd.ctl" is the file that is shared between the two programs and the reason we have problems.
Now open up the clamd.conf file again (mine is `/etc/clamav/clamd.conf`)
Below is illustrated the items in the file we are interested in:

```
LocalSocket /var/run/clamav/clamd.ctl
User clamav
LogFile /var/log/clamav/clamav.log
PidFile /var/run/clamav/clamd.pid
```

```
DatabaseDirectory /var/lib/clamav/
```

We need to edit this file and change:
`User clamav`
to
`User amavis`

Remember, you may be using a different name for your amavisd-new user.
Notice, that in my system, there are 3 directories listed above:
`/var/run/clamav`
`/var/log/clamav`
`/var/lib/clamav`
Now let's change the ownership of the 3 directories shown above (and the files contained therein) so "amavis" can write to them.
Before you do this, be aware, not all installations use a `/var/log/clamav` directory.
If your `LogFile` parameter reads something like `LogFile /var/log/clamav.log`
Then you do not want to change permissions on the entire `/var/log` directory!!!!!
In this case you would only change ownership of the FILE, like so:
`chown amavis:amavis /var/log/clamav.log`
This applies any time the ClamAV file(s) we want to change ownership of are not in a directory specifically created to hold ClamAV files.

```
chown -R amavis:amavis /var/run/clamav
chown -R amavis:amavis /var/lib/clamav
```
and provided you have a separate directory for your log files:
`chown -R amavis:amavis /var/log/clamav`

The virus definition database update program "freshclam" has a configuration file that also needs to be modified.
Mine is called `/etc/clamav/freshclam.conf`
Open this file in your editor. The items we are interested in are:

```
DatabaseOwner clamav
UpdateLogFile /var/log/clamav/freshclam.log
```

Change the DatabaseOwner to **amavis** (or whatever your amavis user is named) and make a note of the location of the log file.
As mentioned above, if `freshclam.log` is not in its own `clamav` directory then only change ownership of the `freshclam.log` file, not the entire directory. In our case, we already changed the ownership of the `/var/log/clamav` directory and all it's contents, so we don't have any more to do here. Your system may differ, so you may need to change ownership.

On my Debian system there are two more files that have to be modified. They are the files that control the maintenance of our log files. You will not necessarily have these files on your system. Our log files get "rotated" by the "logrotate" program each week and these files, if left unchanged, will assign "clamav" as the owner of any new log files it creates. If it does this, we will not be able to write to them. Not a good thing.

These files, on my Debian system are:
`/etc/logrotate.d/clamav-daemon` (controls the clamav.log)
and
`/etc/logrotate.d/clamav-freshclam` (controls the freshclam.log)

The interesting parts of `/etc/logrotate.d/clamav-daemon` on my system are:

```
 create 640  clamav adm
```

```
    /etc/init.d/clamav-daemon reload > /dev/null
```

Edit this file and change:
`create 640 clamav adm`
to
`create 640 **amavis** adm`

Also shown above is how the clamav-daemon is shutdown and restarted.
(`/etc/init.d/clamav-daemon reload`)
Handy to know.

We need to do the same thing with `/etc/logrotate.d/clamav-freshclam`

```
create 640 clamav adm
    /etc/init.d/clamav-freshclam reload > /dev/null
```

Edit this file and change:
`create 640 clamav adm`
to
`create 640 **amavis** adm`

We should reload clamd with the command we found above (`/etc/init.d/clamav-daemon reload`) in order for the daemon to read it's new configuration. Your system will probably differ here. At any rate, you need to stop and restart the clamd process.

Also do the same for freshclam: (`/etc/init.d/clamav-freshclam reload`)
If there are errors in the configuration, it should tell you.
You will also need to stop and restart (or reload) amavisd-new.
If this is a new computer you are building (not in production yet), I suggest you simply reboot.

FYI: These are my configuration files in their entirety (version 0.90.1):

/etc/clamav/clamd.conf:

```
LocalSocket /var/run/clamav/clamd.ctl
FixStaleSocket true
User amavis # user can be clamav if clamav is a member of amavis group
AllowSupplementaryGroups true
ScanMail true
ScanArchive true
ArchiveMaxRecursion 5
ArchiveMaxFiles 1000
ArchiveMaxFileSize 21M
ArchiveMaxCompressionRatio 250
ArchiveLimitMemoryUsage false
ArchiveBlockEncrypted false
MaxDirectoryRecursion 15
FollowDirectorySymlinks false
FollowFileSymlinks false
ReadTimeout 180
MaxThreads 12
MaxConnectionQueueLength 15
StreamMaxLength 10M
LogSyslog false
LogFacility LOG_LOCAL6
LogClean false
```

```
LogVerbose false
PidFile /var/run/clamav/clamd.pid
DatabaseDirectory /var/lib/clamav
TemporaryDirectory /tmp
SelfCheck 3600
Foreground false
Debug false
ScanPE true
ScanOLE2 true
ScanHTML true
DetectBrokenExecutables false
MailFollowURLs false
ArchiveBlockMax false
ExitOnOOM false
LeaveTemporaryFiles false
AlgorithmicDetection true
ScanELF true
NodalCoreAcceleration false
IdleTimeout 30
MailMaxRecursion 64
PhishingSignatures true
LogFile /var/log/clamav/clamav.log
LogTime true
LogFileUnlock false
LogFileMaxSize 0 # only appropriate because I use logrotate
```

/etc/clamav/freshclam.conf:

```
DatabaseOwner amavis # owner can be clamav if clamav is a member of amavis group
UpdateLogFile /var/log/clamav/freshclam.log
LogVerbose false
LogSyslog false
LogFacility LOG_LOCAL6
LogFileMaxSize 0 # only appropriate because I use logrotate
Foreground false
Debug false
MaxAttempts 5
DatabaseDirectory /var/lib/clamav/
DNSDatabaseInfo current.cvd.clamav.net
AllowSupplementaryGroups true
PidFile /var/run/clamav/freshclam.pid
ConnectTimeout 30
ReceiveTimeout 30
ScriptedUpdates yes
NotifyClamd /etc/clamav/clamd.conf
DatabaseMirror db.local.clamav.net
DatabaseMirror database.clamav.net
DatabaseMirror db.us.clamav.net
```

/etc/logrotate.d/clamav-daemon:

```
/var/log/clamav/clamav.log {
     rotate 12
     weekly
     compress
     delaycompress
     create 640  amavis adm
     postrotate
     /etc/init.d/clamav-daemon reload-log > /dev/null
     endscript
```

```
}
```

/etc/logrotate.d/clamav-freshclam:

```
/var/log/clamav/freshclam.log {
     rotate 12
     weekly
     compress
     delaycompress
     create 640 amavis adm
     postrotate
     /etc/init.d/clamav-freshclam reload-log > /dev/null
     endscript
     }
```

The /etc/init.d/clamav-daemon and /etc/init.d/clamav-freshclam startup scripts are specific to Debian.
