Qmail

an internet mail transport agent

or

Qmail – fast, secure, reliable. Pick any three!

Peter Samuel

Peter.Samuel@uniq.com.au

Presented at the AUUG Winter Conference

15 September 1998

Definitions

MUA

Mail user agent. Allows an end user to read incoming mail, respond to this mail and compose new messages.

MTA

Mail transport agent. Responsible for transporting electronic mail from one location to another.

MDA

Mail delivery agent. Responsible for delivering an electronic mail message to its final destination. May be part of the same software suite as the MTA.

Message components

ENVELOPE

Name and address of both the message originator and message recipient(s). Used during the transaction between MTAs.

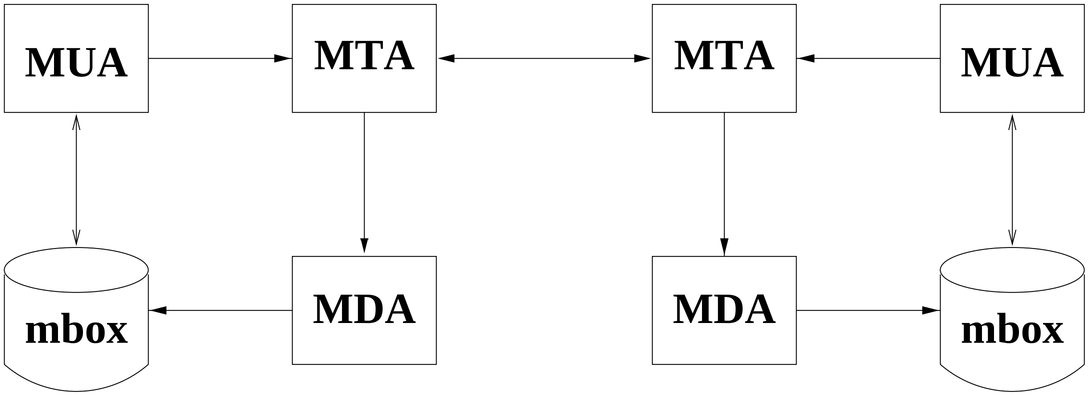
HEADERS

Name and address of both the message originator and message recipient(s) plus extra details such as the date, subject and delivery details.

BODY

Message data.

Simplified mail transaction



The user composes a message using an **MUA.**

The **MUA** gives it to the **MTA** to arrange delivery.

If the message is local, the **MTA** gives it to the local **MDA** which delivers it to the local mailbox where it is available for reading by the **MUA.**

If the message is for a remote recipient, the **MTA** gives it to a remote **MTA** which in turn gives it to that system's **MDA** which delivers it to the remote mailbox.

Qmail History[1]

Dan Bernstein began developing qmail in 1995 in an attempt to avoid the security problems and other deficiencies he saw in sendmail.

The first beta version was 0.70 released in January 1996.

The first production version – 1.00 – was released in February 1997.

Version 1.03 was released in June 1998.

It should compile and run on almost any well behaved Unix system.

It is penetrating the sendmail market. Actual details are hard to determine but judging by the mailing list traffic, many people are either using qmail or want to know more.

[1] This section has been summarised from the qmail distribution files

Qmail security

Segmented architecture. Each stage of the message processing is handled by a different program. Each program runs under a different user ID. The qmail users don’t even trust each other!

The only setuid program is qmail-queue and it runs as the user qmailq.

The only program run by root is the mail delivery agent qmail–lspawn.

qmail–lspawn **ALWAYS** changes its effective user id to the recipient **BEFORE** running any user specified program.

qmail **NEVER** delivers to a user whose UID is zero!

qmail will defer delivery if either the user’s home directory or .qmail file is group and/or world writable.

Qmail security

Parsing is cut down to a bare minimum. This avoids (or at the very least minimises) the risk of introducing a security hole with a configuration error.

The standard C library is avoided – particularly *stdio*.

"Keep It Simple". qmail has only one simple forwarding mechanism that handles forwarding, aliasing and mailing lists, instead of a separate mechanism for each. By default, qmail offers a single delivery mode instead of a selection.

Size comparisons

Number and size of *.c and *.h files[1]

	lines	words	chars	files
qmail-1.03	16617	44780	395243	279
sendmail-8.9.1	55059	179376	1229121	54
zmailer-2.2e10	57595	205524	1423624	227
smail-3.2	62331	246140	1701112	151
exim-2.02	70102	283295	2172786	128

qmail has the smallest code base of the major MTA products. It also has the largest number of files. The obvious assumption that can be made is that qmail adheres to the *"Keep It Simple"* principle.

[1] No attempt was made to strip out comments

Maildir format

Maildirs are essentially incorruptible storage areas for mail messages. They avoid the potential corruption problems of *mbox* and *mh* formats by removing the need for any file locking.

Maildir format stores each message in a separate file in a given directory:

./tmp/ – the delivery agent creates a file called *./tmp/time.pid.hostname* where *time* is the number of seconds since midnight January 1 1970 GMT. The incoming mail message is stored in this file. If the file already exists, the delivery agent waits 2 seconds and tries again.

./new/ – successfully delivered messages are moved to *./new/time.pid.hostname.* The modification time of the file is the delivery date.

./cur/ – messages read by a mail reader are moved to *./cur/time.pid.hostname.*

Aliasing and .qmail files

Users control all addresses of the form *user–anything*.

Addresses which do not map to user names are handled by the special alias user.

Arbitrary addresses can be assigned to specific users using the qmail–users mechanism.

Delivery instructions are handled by .*qmail–ext* files where *ext* is an arbitrary extension.

If my user name is psamuel, I can create a *~/.qmail–jokes* file so that mail sent to psamuel-jokes will be handled by that file.

qmail provides a number of useful environment variables for use in .*qmail* files.

Aliasing and .qmail files

.qmail files can contain any combination of addresses and Unix programs. Each line is a separate delivery instruction.

Exit codes of Unix commands govern further processing of the *.qmail* file.

0	successful delivery
99	successful delivery, ignore all further delivery instructions
100	permanent delivery failure (hard error)
111	transient delivery failure, try again later (soft error)

Example .qmail file[1]

[1] This page is intentionally left blank to illustrate the fact that a .qmail file is not mandatory.

Example .qmail–ext file

A zero byte *.qmail* file instructs qmail–local to deliver the message to the user’s default mailbox.

```
–rw–r—r—  1psamuel    0 Oct 29 18:26 .qmail–ext
```

Example .qmail file

```
# This file is not empty but does nothing.
# Therefore the message will be successfully
# delivered to nowhere.
```

Example .qmail–jokes file

```
# Save messages from the jokes list in a separate mbox.
# Incoming mail will be addressed to psamuel-jokes.
/home/psamuel/mail/JOKES
```

Example .qmail file

```
# My vacation .qmail file

# Completely ignore the message if the sender is
# ddts@x.com. 99 says that the message was delivered
# successfully but do not attempt further delivery
# instructions in the .qmail file. 0 says delivery was
# successful and to continue processing the .qmail file.
| [ "$SENDER" = "ddts@x.com" ] && exit 99 || exit 0

# Run other messages through vacation. preline inserts
# a UUCP style "From user" header, needed by vacation.
| preline /usr/bin/vacation -j psamuel

# Send a copy to my ISP account. The & is only
# necessary if the address does not begin with
# an alphanumeric character.
&peter@isp.com.au

# Save a local copy - most important otherwise
# there will be no local delivery. Also save a copy
# to a maildir so that I can quickly scan individual
# messages when I return. The trailing slash indicates
# a maildir instead of an mbox.
/home/psamuel/Mailbox
/home/psamuel/Maildir/
```

Mandatory qmail control file

qmail requires a single control file for its operation:

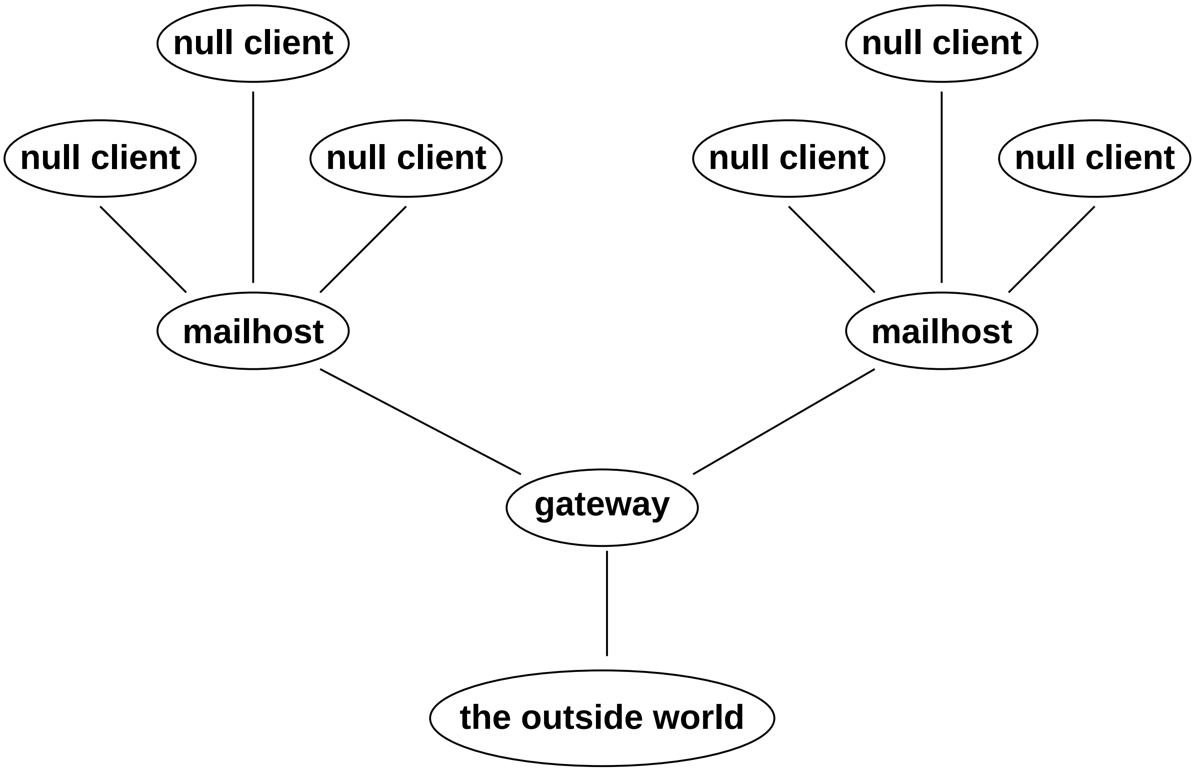
```
/var/qmail/control/me
```

This file contains the fully qualified host name of the current host. If this file is missing, the major qmail programs – qmail-smtpd, qmail-send and qmail-remote – will refuse to run.

Optional qmail control files

control file	**default value**	**used by**
badmailfrom	(none)	qmail-smtpd
bouncefrom	MAILER-DAEMON	qmail-send
bouncehost	me	qmail-send
concurrencylocal	10	qmail-send
concurrencyremote	20	qmail-send
defaultdomain	me	qmail-inject
defaulthost	me	qmail-inject
databytes	0	qmail-smtpd
doublebouncehost	me	qmail-send
doublebounceto	postmaster	qmail-send
envnoathost	me	qmail-send
helohost	me	qmail-remote
idhost	me	qmail-inject
localiphost	me	qmail-smtpd
locals	me	qmail-send
morercpthosts	(none)	qmail-smtpd
percenthack	(none)	qmail-send
plusdomain	me	qmail-inject
qmqpservers	(none)	qmail-qmqpc
queuelifetime	604800	qmail-send
rcpthosts	(none)	qmail-smtpd
smtpgreeting	me	qmail-smtpd
smtproutes	(none)	qmail-remote
timeoutconnect	60	qmail-remote
timeoutremote	1200	qmail-remote
timeoutsmtpd	1200	qmail-smtpd
virtualdomains	(none)	qmail-send

Example mail network



Configuring a null client

A null client is a machine which does **NO** local mail delivery.

ALL outgoing mail is sent to a single mailhost.

/var/qmail/control/locals – an empty file

/var/qmail/control/smtproutes

```
        :mailhost.devel.domain.com
```

If you can’t resolve host names you can use the IP address instead.

/var/qmail/control/smtproutes

```
        :[123.45.67.89]
```

Configuring a null client

If you are using **qmail-1.03** or later you can eliminate **ALL** local mail queueing by using QMQP (Quick Mail Queueing Protocol) instead of SMTP.

On the client

/var/qmail/bin/qmail-queue – a symbolic link to **/var/qmail/bin/qmail-qmqpc**

/var/qmail/control/locals – an empty file

/var/qmail/control/qmqpservers

```
123.45.67.89
```

On the server

/etc/inetd.conf – all on one line

```
qmqp stream tcp nowait qmaild /pkgs/bin/tcpd
   /var/qmail/bin/tcp-env /var/qmail/bin/qmail-qmqpd
```

/etc/services

```
qmqp   628/tcp       mail
```

Configuring a mailhost

A mailhost is a machine that will perform all the local deliveries for a site.

It accepts mail from null clients and from external sources.

It sends **ALL** outgoing mail to a single mail gateway.

/var/qmail/control/locals

```
mailhost.devel.domain.com
devel.domain.com
```

/var/qmail/control/smtproutes

```
:gateway.domain.com
```

/etc/inetd.conf – all on one line

```
smtp stream tcp nowait qmaild /pkgs/bin/tcpd
   /var/qmail/bin/tcp-env /var/qmail/bin/qmail-smtpd
```

Configuring a gateway

A gateway is a machine that sends incoming mail to the relevant internal mailhost.

It accepts mail from mail hosts and from external sources.

It sends **ALL** outgoing mail to the relevant MX host.

/var/qmail/control/locals

```
gateway.domain.com
domain.com
```

/var/qmail/control/smtproutes

```
devel.domain.com:mailhost.devel.domain.com
sales.domain.com:mailhost.sales.domain.com
```

/etc/inetd.conf – all on one line

```
smtp stream tcp nowait qmaild /pkgs/bin/tcpd
   /var/qmail/bin/tcp-env /var/qmail/bin/qmail-smtpd
```

Host masquerading

"How do I set up host masquerading?"

/var/qmail/control/defaultdomain

```
devel.domain.com
```

Unqualified From: addresses will now have devel.domain.com appended.

Users can also modify their From: headers by using environment variables.

```
export QMAILHOST=some.domain
export QMAILNAME=Peter Samuel
export QMAILUSER=Peter.Samuel
export QMAILINJECT=f
```

This becomes:

```
From: Peter Samuel <Peter.Samuel@some.domain>
```

Virtual domains

/var/qmail/control/virtualdomains

```
cardassia.empire:garok
ferengi.alliance:quark
klingon.empire:worf
starfleet.fed:kirk
```

Mail for user@klingon.empire will be rewritten as worf–user@klingon.empire and delivered locally to the user worf according to the instructions in ~worf/.qmail–user or ~worf/.qmail-default.

Handling SPAM

SPAM is another name for Unsolicited Commercial Email or UCE. Qmail has a number of mechanisms for limiting the amount of SPAM received.

The default configuration does not support the *"percent hack"* address convention. An address of the form

user%domain1@domain2

will **NOT** be forwarded to *domain1*. The *"percent hack"* can be enabled for given domains using a control file.

Qmail can be configured to prevent mail relaying by specifying valid incoming domains in */var/qmail/control/rcpthosts*. This can be overridden by conditionally setting the **$RELAYCLIENT** environment variable for each invocation of qmail-smtpd using tcp_wrappers or tcpserver.

Mail from individual users or complete domains can be banned by specifying the address or domain in */var/qmail/control/badmailfrom*.

Handling SPAM

Specific IP address connection control can be achieved using tcp_wrappers (**ftp://ftp.win.tue.nl/pub/security)** or tcpserver, part of Dan’s ucspi-tcp package (**ftp://koobera.math.uic.edu/www/ucspi-tcp.html).**

Dan has released rblsmtpd which blocks spam according to the *Real Time Blacklist*. See **ftp://koobera.math.uic.edu/www/rblsmtpd.html** for details on rblsmtpd. See **http://maps.vix.com/rbl** for more details on the *Real Time Blacklist*.

There are a number of anti-spam patches for qmail–smtpd available from **http://www.qmail.org**

If SPAM control is a serious issue, you may be better off using a different SMTP daemon such as smtpd from **http://www.obtuse.com/smtpd.html.**

For other SPAM solutions, see the offerings at **http://spam.abuse.net/spam/.**

Disadvantages of using qmail

Qmail is an internet mailer. It will only talk **SMTP** to other MTAs. (It can be configured to use a more efficient protocol when talking to other qmail systems).

It will **ONLY** handle addresses which are in the form of user@domain. UUCP style addresses such as domain!user are definitely **NOT** supported. However transport through UUCP is easy to set up for both incoming and outgoing messages.

It makes every effort to conform to the existing standards and RFCs. Non conforming systems may have difficulty in communicating with qmail–smtpd. Some patches are available to alleviate this problem. See **http://www.qmail.org.**

Header re–writing is minimal. However inserting your own rewrite program is easy.

Disadvantages of using qmail

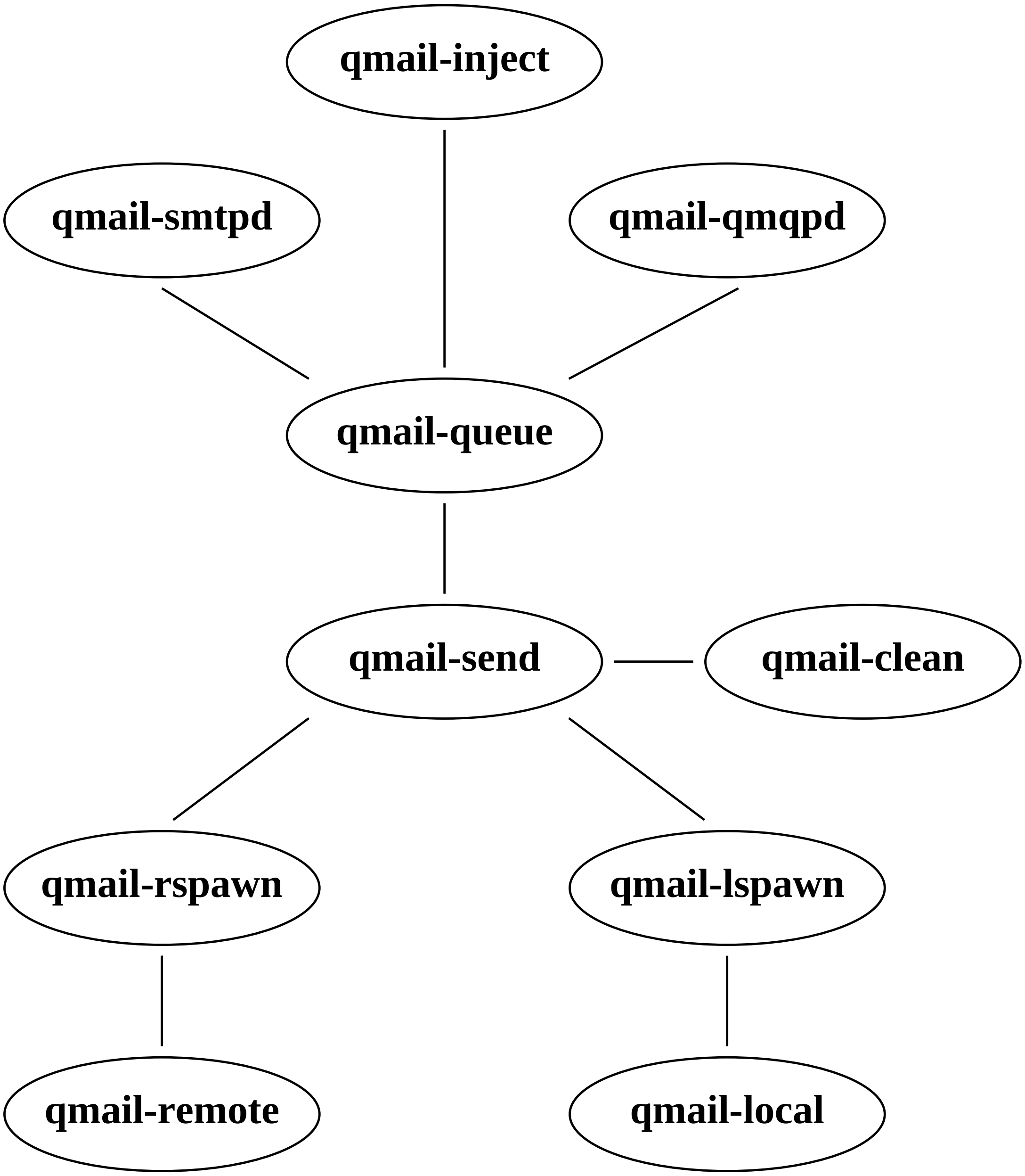
The author is resistant to change unless it can be shown that the change is absolutely necessary. He is known as being quite arrogant and dogmatic. Unfortunately (or fortunately depending on your point of view) he's usually right. 🙂

Qmail requires a completely different mind set than sendmail. It does take some time to begin to think in a qmail fashion – especially if you are used to thinking in a sendmail fashion.

Distributing modified versions of qmail – including ports – require the author's explicit permission. Dan wants to make certain that distributions do not unintentionally compromise performance, operation and security.

Precompiled binary versions of qmail may be distributed provided the package conforms to the guidelines set down by the author. See **ftp://koobera.math.uic.edu/www/qmail.html** for more details.

How qmail works



How qmail works

Incoming mail arrives via either an SMTP connection to qmail–smtpd or via indirect queue injection from qmail–inject.

qmail–smtpd or qmail–inject calls qmail–queue to place the message in the queue.

qmail–queue splits the message into three files, the message itself and two copies of the envelope details. It then signals qmail–send to begin delivery.

If the message is for a local recipient, qmail–send tells qmail–lspawn to deliver the message. qmail–lspawn calls qmail–local to deliver the message. qmail–local handles alias expansion and if necessary calls qmail–queue to place a new message in the queue. qmail–local finally delivers the message to the user’s home directory.

If the message is for a remote recipient, qmail–send tells qmail–rspawn to deliver the message. qmail–rspawn calls qmail–remote to deliver the message to a remote MTA.

Message pre–processing

qmail-queue receives incoming messages and places them in the queue as follows:

/var/qmail/queue/pid/*tempfile*

Temporary file with unique name.

/var/qmail/queue/mess/0..22/*nnnnn*

Hard link to /var/qmail/queue/pid/*tempfile*. Filename is the same as its inode. Contains message headers and body. /var/qmail/queue/pid/*tempfile* is removed.

/var/qmail/queue/intd/*nnnnn*

Message envelope in creation.

/var/qmail/queue/todo/*nnnnn*

Completed message envelope.

/var/qmail/queue/lock/trigger

Named pipe for communication with qmail-send.

The message is now completely pre–processed and is awaiting delivery by qmail-send.

Message processing

qmail-send processes queued messages as follows:

/var/qmail/queue/info/0..22/*nnnnn*

Envelope From: address. Created from /var/qmail/queue/todo/*nnnnn*.

/var/qmail/queue/local/0..22/*nnnnn*

Envelope To: address if recipient is local. Created from /var/qmail/queue/todo/*nnnnn*. qmail-lspawn is notified to deliver the message locally. When all local deliveries have been performed, qmail-send deletes the file.

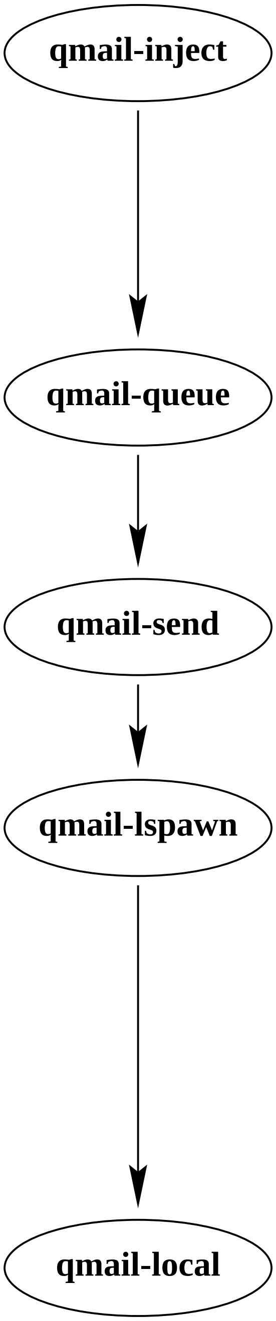
/var/qmail/queue/remote/0..22/*nnnnn*

Envelope To: address if recipient is remote. Created from /var/qmail/queue/todo/*nnnnn*. qmail-rspawn is notified to deliver the message remotely. When all remote deliveries have been performed, qmail-send deletes the file.

/var/qmail/queue/bounce/0..22/*nnnnn*

If a delivery has a permanent failure, this file is created and/or appended to. When all deliveries are finished a bounce message created from this file and /var/qmail/queue/mess/0..22/*nnnnn* is injected into the queue.

Local to local delivery



Original message:

To: fred

Hi.

Fill in the complete envelope and header:

```
(envelope) from joe@heaven.af.mil \
    to fred@heaven.af.mil
From: joe@heaven.af.mil
To: fred@heaven.af.mil

Hi.
```

Store message safely on disk.
Trigger qmail-send.

Look at envelope recipient, fred@heaven.af.mil.
Is heaven.af.mil in locals? Yes.
Deliver locally to fred@heaven.af.mil.

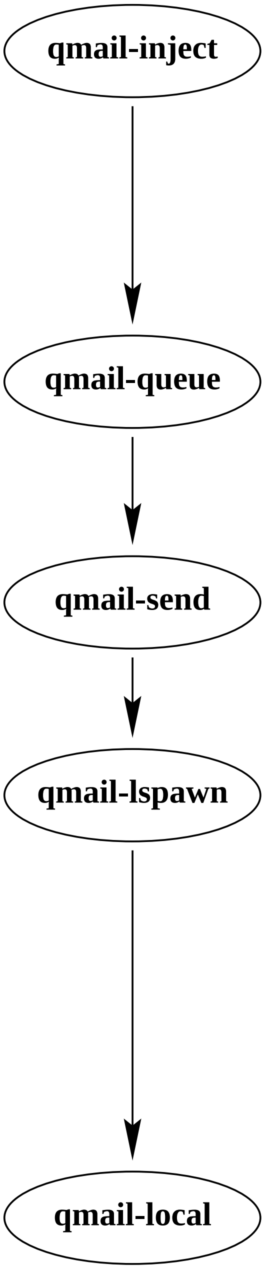
qmail-lspawn ./Mailbox

Look at mailbox name, fred.
Is fred listed in qmail-users? No.
Is there a fred account? Yes.
Is fred’s uid nonzero? Yes.
Is ~fred visible to the qmailp user? Yes.
Is ~fred owned by fred? Yes.
Give control of the message to fred.
Run qmail-local.

```
qmail-local fred ~fred fred ’’ ’’ heaven.af.mil \
    joe@heaven.af.mil ./Mailbox
```

Does ~fred/.qmail exist? No.
Write message to ./Mailbox in mbox format.

Local to local extension delivery



Original message:

```
To: fred-sos
Hi.
```

Fill in the complete envelope and header:

```
(envelope) from joe@heaven.af.mil \
    to fred-sos@heaven.af.mil
From: joe@heaven.af.mil
To: fred-sos@heaven.af.mil

Hi.
```

Store message safely on disk.
Trigger qmail-send.

Look at envelope recipient,
fred-sos@heaven.af.mil.
Is heaven.af.mil in locals? Yes.
Deliver locally to fred-sos@heaven.af.mil.

```
qmail-lspawn ./Mailbox
```

Look at mailbox name, fred-sos.
Is fred-sos listed in qmail-users? No.
Is there a fred-sos account? No.
Is there a fred account?Yes.
Is fred's uid nonzero? Yes.
Is ~fred visible to the qmailp user? Yes.
Is ~fred owned by fred? Yes.
Give control of the message to fred.
Run qmail-local.

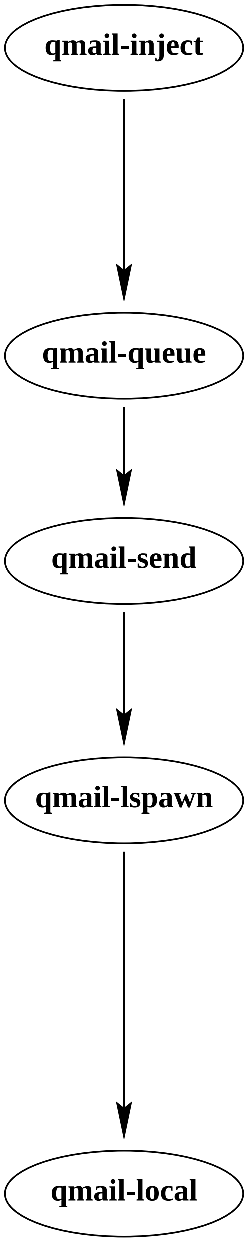
```
qmail-local fred ~fred fred-sos - sos heaven.af.mil \
    joe@heaven.af.mil ./Mailbox
```

Does ~fred/.qmail-sos exist? Yes. "./Extramail".
Write message to ./Extramail in mbox format.

Local to virtual delivery

Original message:

```
To: dude@tommy.gov
Hi.
```



Fill in the complete envelope and header:

```
(envelope) from joe@heaven.af.mil \
    to dude@tommy.gov
From: joe@heaven.af.mil
To: dude@tommy.gov

Hi.
```

Store message safely on disk.
Trigger qmail-send.

Look at envelope recipient, dude@tommy.gov.
Is tommy.gov in locals? No.
Is dude@tommy.gov in virtualdomains? No.
Is tommy.gov in virtualdomains? Yes:
"tommy.gov:fred"
Deliver locally to fred-dude@tommy.gov.

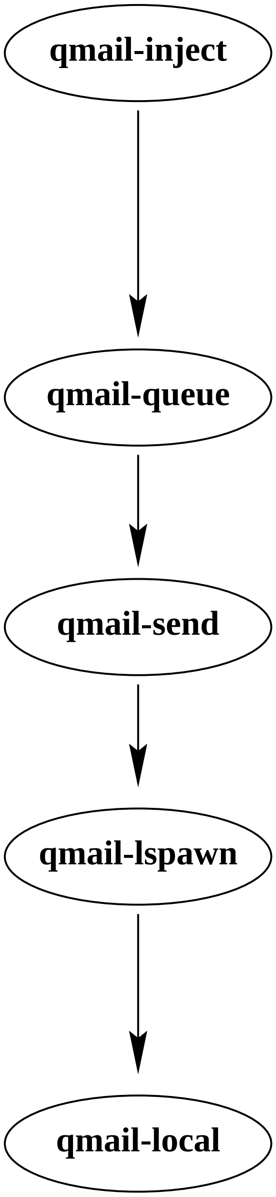
```
qmail-lspawn ./Mailbox
```

Look at mailbox name, fred-dude.
Is fred-dude listed in qmail-users? No.
Is there a fred-dude account? No.
Is there a fred account?Yes.
Is fred’s uid nonzero? Yes.
Is ~fred visible to the qmailp user? Yes.
Is ~fred owned by fred? Yes.
Give control of the message to fred.
Run qmail-local.

```
qmail-local fred ~fred fred-dude - dude tommy.gov \
    joe@heaven.af.mil ./Mailbox
```

Does ~fred/.qmail-dude exist? No.
Does ~fred/.qmail-default exist? Yes: "./Mail.tommy".
Write message to ./Mail.tommy in mbox format.

Local to local alias delivery



Original message:

```
To: help
Hi.
```

Fill in the complete envelope and header:

```
(envelope) from joe@heaven.af.mil \
    to help@heaven.af.mil
From: joe@heaven.af.mil
To: help@heaven.af.mil

Hi.
```

Store message safely on disk.
Trigger qmail-send.

Look at envelope recipient, help@heaven.af.mil.
Is heaven.af.mil in locals? Yes.
Deliver locally to help@heaven.af.mil.

```
qmail-lspawn ./Mailbox
```

Look at mailbox name, help.
Is help listed in qmail-users? No.
Is there a help account? No.
Give control of the message to alias.
Run qmail-local.

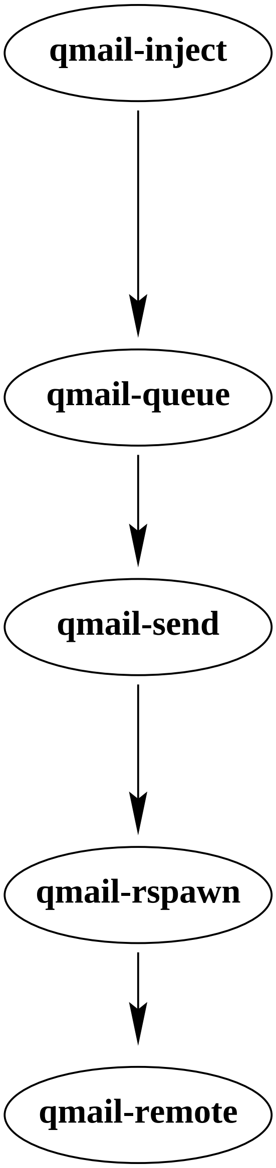
```
qmail-local alias ~alias help - help heaven.af.mil \
    joe@heaven.af.mil ./Mailbox
```

Does ~alias/.qmail-help exist? Yes: "john".
Forward message to john.
(delivery is now a local to local delivery)

Local to remote delivery

Original message:

```
To: bill@irs.gov
Hi.
```



qmail-inject

Fill in the complete envelope and header:

```
(envelope) from joe@heaven.af.mil \
    to bill@irs.gov
From: joe@heaven.af.mil
To: bill@irs.gov

Hi.
```

qmail-queue

Store message safely on disk.
Trigger qmail-send.

qmail-send

Look at envelope recipient, bill@irs.gov.
Is irs.gov in locals? No.
s bill@irs.gov in virtualdomains? No.
Is irs.gov in virtualdomains? No.
Is .gov in virtualdomains? No.
Deliver remotely to bill@irs.gov.

qmail-rspawn

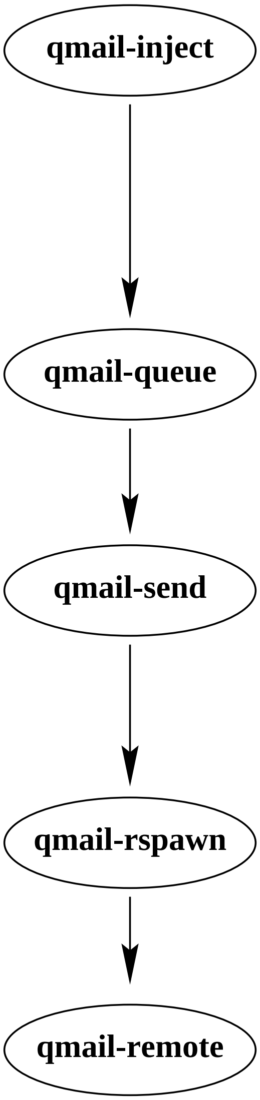
Run qmail-remote.

qmail-remote

Look at host name, irs.gov.
Is irs.gov listed in smtproutes? No.
Look up DNS MX/A for irs.gov and
connect to it by SMTP.

```
MAIL FROM: <joe@heaven.af.mil>
RCPT TO: <bill@irs.gov>
```

Null client delivery



Original message:

To: bill@irs.gov
Hi.

Fill in the complete envelope and header:

**(envelope) from joe@heaven.af.mil \\
to bill@irs.gov
From: joe@heaven.af.mil
To: bill@irs.gov**

Hi.

**Store message safely on disk.
Trigger qmail-send.**

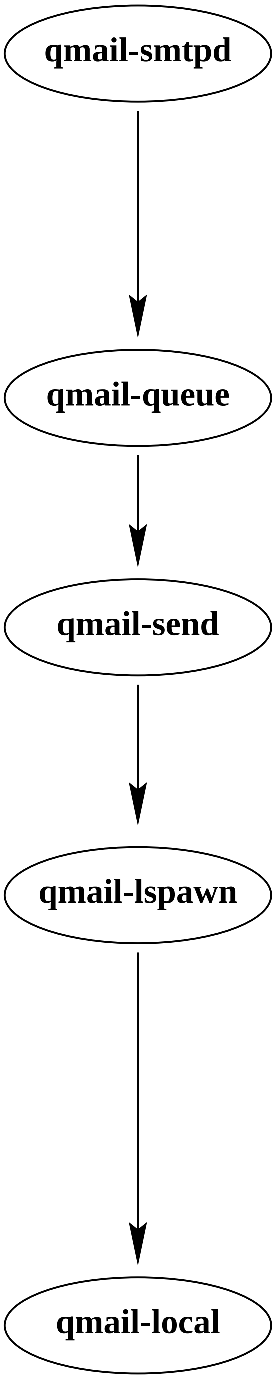
**Look at envelope recipient, bill@irs.gov.
Is irs.gov in locals? No.
Is bill@irs.gov in virtualdomains? No.
Is irs.gov in virtualdomains? No.
Is .gov in virtualdomains? No.
Deliver remotely to bill@irs.gov.**

Run qmail-remote.

**Look at host name, irs.gov.
Is irs.gov listed in smtproutes? Yes:
"bigbang.af.mil".
Look up DNS A for bigbang.af.mil and
connect to it by SMTP.**

**MAIL FROM: <joe@heaven.af.mil>
RCPT TO: <bill@irs.gov>**

Remote to local delivery



qmail-smtpd

Receive message by SMTP from another host:

MAIL FROM: <bill@irs.gov>
RCPT TO: <joe@heaven.af.mil>

Is $RELAYCLIENT set? No.
Is heaven.af.mil in rcpthosts? Yes.
Accept RCPT.

qmail-queue

Store message safely on disk.
Trigger qmail-send.

qmail-send

Look at envelope recipient, joe@heaven.af.mil.
Is heaven.af.mil in locals? Yes.
Deliver locally to joe@heaven.af.mil.

qmail-lspawn

qmail-lspawn ./Mailbox

Look at mailbox name, joe.
Is joe listed in qmail-users? No.
Is there a joe account? Yes.
Is joe’s uid nonzero? Yes.
Is ~joe visible to the qmailp user? Yes.
Is ~joe owned by joe? Yes.
Give control of the message to joe.
Run qmail-local.

qmail-local

**qmail-local joe ~joe joe ’’ ’’ heaven.af.mil **
bill@irs.gov ./Mailbox

Does ~joe/.qmail exist? No.
Write message to ./Mailbox in mbox format.

Unauthorised relaying

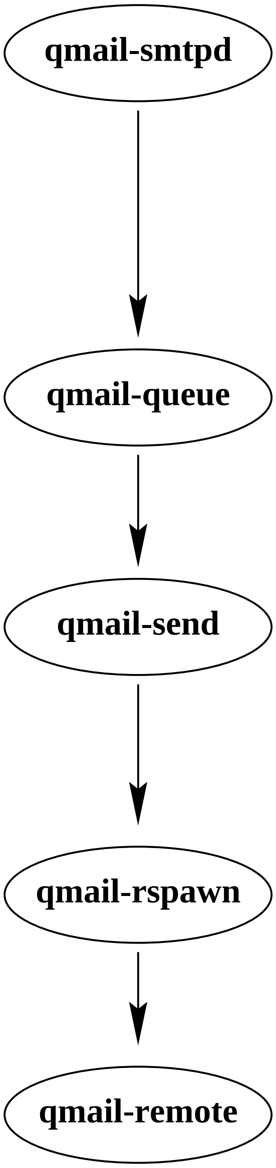
qmail-smtpd

Receive message by SMTP from another host:

```
MAIL FROM: <spammer@aol.com>
RCPT TO: <bill@irs.gov>
```

Is $RELAYCLIENT set? No.
Is irs.gov in rcpthosts? No.
Reject RCPT.

Authorised relaying



qmail-smtpd

Receive message by SMTP from another host:

```
MAIL FROM: <joe@heaven.af.mil>
RCPT TO: <bill@irs.gov>
```

Is $RELAYCLIENT set? Yes.
Accept RCPT.

qmail-queue

Store message safely on disk.
Trigger qmail-send.

qmail-send

Look at envelope recipient, bill@irs.gov.
Is irs.gov in locals? No.
Is bill@irs.gov in virtualdomains? No.
Is irs.gov in virtualdomains? No.
Is .gov in virtualdomains? No.
Deliver remotely to bill@irs.gov.

qmail-rspawn

Run qmail-remote.

qmail-remote

Look at host name, irs.gov.
Is irs.gov listed in smtproutes? No.
Look up DNS MX/A for irs.gov and
connect to it by SMTP.

```
MAIL FROM: <joe@heaven.af.mil>
RCPT TO: <bill@irs.gov>
```

Starting qmail

The qmail distribution provides a number of startup examples in /var/qmail/boot/*. The basic mechanism is

```
env - PATH="/var/qmail/bin:$PATH" \
   qmail-start [ aliasempty [ logger arg ... ] ]
```

The aliasempty argument is given to qmail-lspawn. It can be a simple mbox filename such as ./Mailbox or a maildir directory name such as ./Maildir/.[1]

The aliasempty argument may also be a different delivery program such as /bin/mail or procmail for example.

qmail's activity records are sent through the program specified by the logger argument – typically qmail's splogger.

The start up environment is eventually passed to qmail-local so you may need to sanitise the environment before starting qmail.

You may also wish to place limits on qmail's resource usage using appropriate ulimit settings.

[1] The trailing slash is vital. It signifies that the delivery destination is a maildir directory rather than an mbox file called Maildir.

Migrating from sendmail

By default qmail will deliver to each user’s home directory. If you wish to preserve your /var/spool/mail/user mail boxes, instruct qmail to deliver mail using a different MDA:

```
qmail-start \
   ’|preline -f /bin/mail -r \
   "${SENDER:-MAILER-DAEMON}" -d $USER’ \
   splogger qmail
```

Examples of different MDA configurations are available in /var/qmail/boot/*.

/etc/aliases compatibility

If you don't want to move away from your existing /etc/aliases file you can use the fastforward package.

Create a hash table of alias definitions by running

```
newaliases
```
[1]

If you have :include references, create hash tables for each of these include files by running

```
newinclude /path/to/include/file
```

Enable the alias database(s) by creating ~alias/.qmail-default which contains

```
| fastforward -d /etc/aliases.cdb
```

fastforward will automatically read the contents of any :include references made in /etc/aliases provided you have run the newinclude command.

fastforward can also be used to maintain separate alias databases for each supported virtual domain.

[1] This is the fastforward version of newaliases and is not to be confused with the traditional sendmail version.

.forward compatibility

If you don’t want to move away from your existing .forward files you can use the dot-forward package.

Start qmail by specifying .forward compatibility

```
qmail-start '| dot-forward .forward[1]
./Mailbox' splogger qmail
```

If a user has a .forward file, qmail will obey the instructions in that file and no further delivery will take place. If the .forward file does not exist, qmail will deliver to the user’s default mail box – specified above as ./Mailbox.

If you don’t wish to provide .forward compatibility to all users, individual users can take advantage of dot-forward by specifying dot-forward delivery in their own .qmail files.

```
| dot-forward .forward
./Mailbox
```

[1] The newline here is vital. The dot-forward delivery instruction is separate from the ./Mailbox delivery instruction.

Mailing list management

Majordomo version 1.9*x* needs some modification to work with qmail. Version 2.*x* should support qmail *"out of the box"*.

ftp://ftp.eyrie.org/pub/software/majordomo/mjqmail is an excellent FAQ on how to modify majordomo. It contains pointers to a number of patches and auxiliary programs that allow you to continue using majordomo with qmail.

The basic steps involve

- Apply patches to majordomo
- Eliminate majordomo’s wrapper program
- Create ~majordomo/.qmail-*list* files
- Change majordomo config files to use *list*-owner instead of owner-*list*

Ezmlm – a qmail mailing list manager

Ezmlm is specifically tailored for qmail. Features include:

- secure and simple subscribe/unsubscribe mechanism
- automatic message archiving
- open or closed lists
- smart bounce handling using qmail’s Variable Envelope Return Path (VERP) to reliably determine the recipient address and message number for every incoming bounce message
- automatic unsubscription after 10 days of bounces
- any user can create an ezmlm list
- customisations available through .qmail files

Creating an ezmlm list

To create a psamuel–jokes list

```
ezmlm-make ~/JOKES ~/.qmail-jokes \
   psamuel-jokes my.domain.com
```

Subscribe/unsubscribe requests are sent to

psamuel–jokes–subscribe@my.domain.com
psamuel–jokes–unsubscribe@my.domain.com

Postings are sent to

psamuel–jokes@my.domain.com

List help is sent to

psamuel–jokes–help@my.domain.com

The list owner can be contacted at

psamuel–jokes–owner@my.domain.com

Users send retrieval requests for message 12345 to

psamuel–jokes–get.12345@my.domain.com

Customising an ezmlm list

Disable message archiving

```
mv ~/JOKES/archived ~/JOKES/notarchived
```

Prevent public subscription

```
mv ~/JOKES/public ~/JOKES/notpublic
```

Force a Reply-To: list header

```
echo "Reply-To: psamuel-jokes@my.domain.com" \
    >> ~/JOKES/headeradd
```

Allow closed user group subscription

```
touch ~/JOKES/public
```

and then edit ~/.qmail-jokes-default and insert

```
|domaincheck listname domain1 domain2 ...
```

Domaincheck is an executable of your own design that examines $SENDER and exits zero if the sender’s domain is in one of the listed valid domains. If the sender is not from one of these domains, domaincheck prints an appropriate message to stdout and exits 100 which informs qmail that no further processing is to take place.

Customising an ezmlm list

Further possible customisation include:

- modify ~/JOKES/headerremove to exclude specific headers
- modifying administrative text messages by editing ~/JOKES/text/*
- adding list specific footer messages by modifying ~/.qmail-jokes (all on one line):

  ```
  | cat - /home/psamuel/JOKES/text/footer[1] |
     /pkgs/bin/ezmlm-send '/home/psamuel/JOKES'
  ```

- modifying Subject: headers and adding a footer message by modifying ~/.qmail-jokes (all on one line):

  ```
  | subject-prefix[2] JOKES |
     cat - /home/psamuel/JOKES/text/footer |
     /pkgs/bin/ezmlm-send '/home/psamuel/JOKES'
  ```

- just about anything you can think of to do in a .qmail file

[1] footer is a text file of your own making

[2] subject-prefix is a program of your own design that modifes the Subject: header of the message

Availability and competing products

qmail – version 1.03, (June 1998)

http://www.qmail.org
ftp://koobera.math.uic.edu/www/qmail.html

sendmail – version 8.9.1, (July 1998)

http://www.sendmail.org

exim – version 2.02, (August 1998)

http://www.exim.org

smail – version 3.2, (July 1996, latest beta August 1998)

ftp://ftp.uu.net/networking/mail/smail

zmailer – version 2.2e10, (March 1996)

ftp://ftp.cs.toronto.edu/pub/zmailer

VMailer – Not available, still in alpha testing

http://wzv.win.tue.nl/vmail

Ancillary software

qmail is distributed with a number of ancillary programs.

qmail–qstat – show the number of queued and pre–processed messages in the queue.

qmail–qread – shows the sender and recipient details and delivery status of all pre–processed messages.

qmail–pop3d – a POP3 daemon that supports maildir formats. Requires a checkpassword program.

sendmail wrapper – allows local MUA programs to be used unchanged. They can still compose mail and deliver it via */usr/lib/sendmail –t*. Calls qmail–inject.

Ancillary software

datemail – similar interface to the sendmail wrapper. Sets Date: header in localtime instead of GMT for those MUAs that do not set the header. Calls qmail–inject.

preline – inserts a UUCP style From user header before calling an external program.

forward – forwards mail messages to the specified list of addresses. List can be constructed *"on the fly"*.

And many more.

Additional software

Dan Bernstein has written a number of other packages that can be used with qmail.

qmailanalog – analyses log files and generates mail statistics.

ezmlm – mailing list manager specifically for qmail. Supports automatic subscription, archiving and retrieval of messages.

checkpassword – required by qmail–pop3d. Originally written by Russ Nelson.

fastforward – sendmail like alias clone for sites that don't wish to move away from */etc/aliases* files.

dot-forward – sendmail like *.forward* clone for sites that don't wish to move away from *.forward* files.

Additional software

serialmail – allows batch processing of mail transport over slow/intermittent serial links.

rblsmtpd – provides support for excluding mail from sites listed in the *Real Time Blacklist*. It can be used with any SMTP daemon. See **http://maps.vix.com/rbl** for more information on the *Real Time Blacklist*.

Third party software

There are a host of other third party packages available from the qmail web site – **http://www.qmail.org.**

Extensions to ezmlm.

Patches to provide maildir support for MUAs.

Alternative checkpassword programs.

Smart startup/shutdown wrappers.

HTML versions of the manual pages.

qmail patches for better SPAM blocking etc.

Qmail book

John Levine and Russell Nelson are writing a book, to be published in 1998 by O’Reilly & Associates.

Commercial Support

At least three organisations are willing to provide commercial support for qmail.

Mira Networking. (Australia) Support is available over the Internet and Mira accepts Visa/MC/Amex/Diners.

http://www.mira.net/qmail.html

Crynwr Software. (USA) Support is available on-site, by 800 number, or over the Internet. Crynwr accepts Visa/MC/Amex/purchase orders.

http://www.crynwr.com/qmail.html

Inter7 Internet Technologies, Inc. (USA) Support is available on-site, by 800 number or over the internet. Inter7 accepts purchase orders.

http://www.inter7.com/qmail

The future

Zero administration for null clients. Replace SMTP with QMQP for communication between null clients and central mail host. A full implementation of QMQP ships with qmail version 1.03. Dan plans to release a mini qmail package with QMQP support enabled by default.

Faster installation. Provide precompiled packages for popular operating systems.

Split log analysis. Improvements to qmailanalog. The next version will be designed to run periodically, keeping crucial information in a table on disk.

[Speed] Wide–area QMTP support. Use QMTP between qmail hosts to cut down on SMTP latency. Requires encoding the QMTP availability in the hosts MX record. See RFCMXPS and RFCQMTP in the qmail distribution for more details.

The future

[Speed] Asynchronous compressed journaling. Reduce qmail’s disk I/O by feeding new mail through a separate journaling process that saves messages in compressed form; qmail–send will rebuild the queue from the compressed journal when it starts.

[Speed] Local–host DNS lookups. A future version of qmail will support an alternate, much smaller DNS library that talks to a nameserver on the same host.

Dynamic subscription agents Why should users have to deal with dozens of different mailing list subscription mechanisms? The new **dynasub** package will accept subscription requests from local users and negotiate subscriptions with remote mailing lists. It will automatically set up a local sublist for each remote list, to speed delivery and protect user privacy.