

*Qmail*

*an internet mail transport agent*

*or*

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

Peter Samuel

Presented to the Sydney Linux Users Group



24 April 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. Used during the transaction between MTAs.

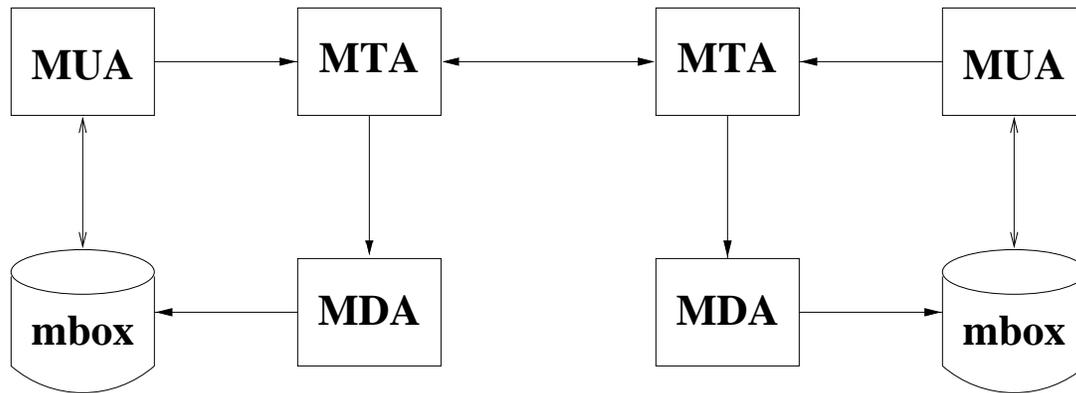
## **HEADERS**

Name and address of both the message originator and message recipient 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<sup>1</sup>

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 1.01 was released in July 1997.

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.

---

<sup>1</sup>This section has been summarised from the qmail-1.01 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 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 the root user!

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. qmail offers a single delivery mode instead of a selection.

## Size comparisons

Number and size of \*.c and \*.h files

	<b>lines</b>	<b>words</b>	<b>chars</b>	<b>files</b>
qmail-1.01	16028	44331	370123	288
sendmail-8.8.8	52830	179608	1218116	53
zmailer-2.2e10	57595	205524	1423624	227
smail-3.2	62331	246140	1701112	151
exim-1.90	67778	272084	2092351	127

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.

## 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<sup>1</sup>

---

<sup>1</sup>This page is intentionally left blank

## Example .qmail file

A zero byte *.qmail* file instructs `qmail-local` to deliver the message to the user's default mailbox.

```
-rw-r--r-- 1 psamuel  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 file**

```
# Save messages from the jokes list in a separate mbox.  
/home/psamuel/mail/JOKES
```

## Example .qmail file

# My vacation .qmail file

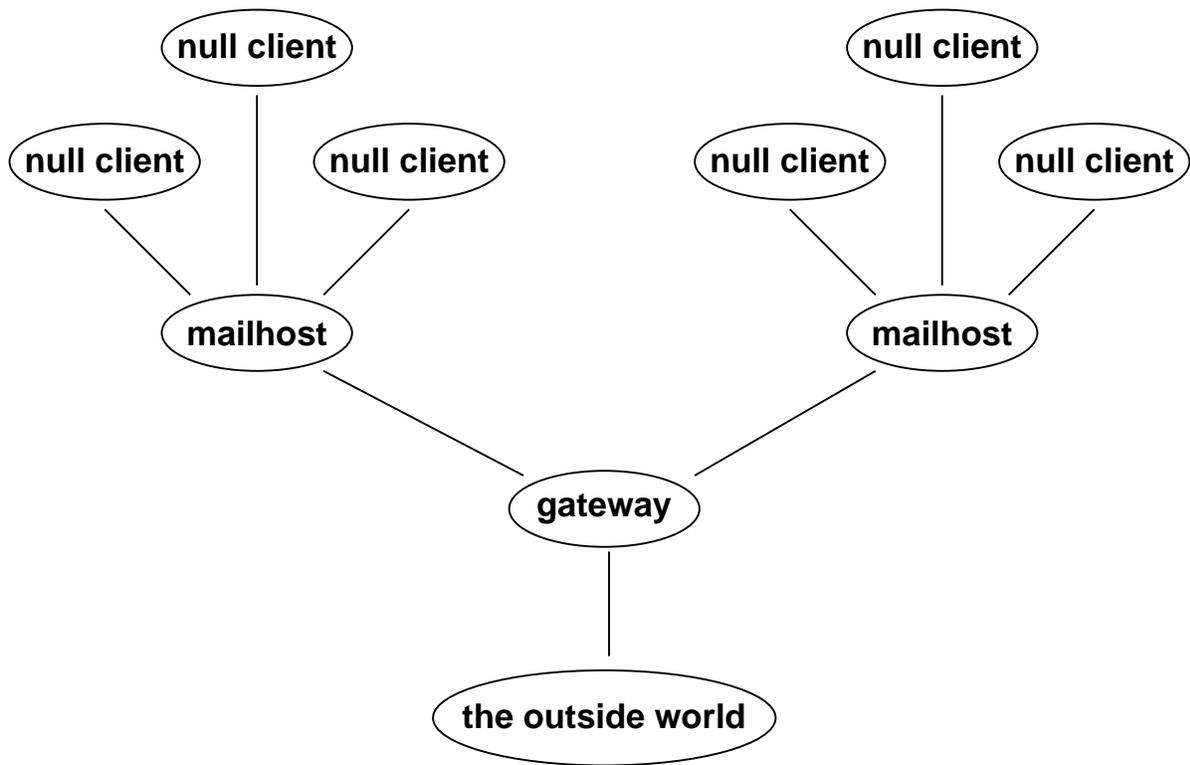
# Completely ignore the message if the sender is  
# ddts@x.com. 99 says that the message was delivered  
# successfully but do not attempt further delivery  
# instructions in the .qmail file. 0 says delivery was  
# successful and to continue processing the .qmail file.  
| [ "\$SENDER" = "ddts@x.com"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

# 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 hostnames you can use the IP address instead.

**/var/qmail/control/smtproutes**

:[123.45.67.89]

## 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
```

## Configuring a gateway

A gateway is a machine that will send 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
```

## 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.

# Virtualdomains

## **`/var/qmail/control/virtualdomains`**

cardassia.empire:garok

ferengi.alliance:quark

klinton.empire:worf

starfleet.fed:kirk

Mail for `user@klinton.empire` will be rewritten as `worf-user@klinton.empire` and delivered locally.

## 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>).

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 **NOT** handle addresses which are not 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.

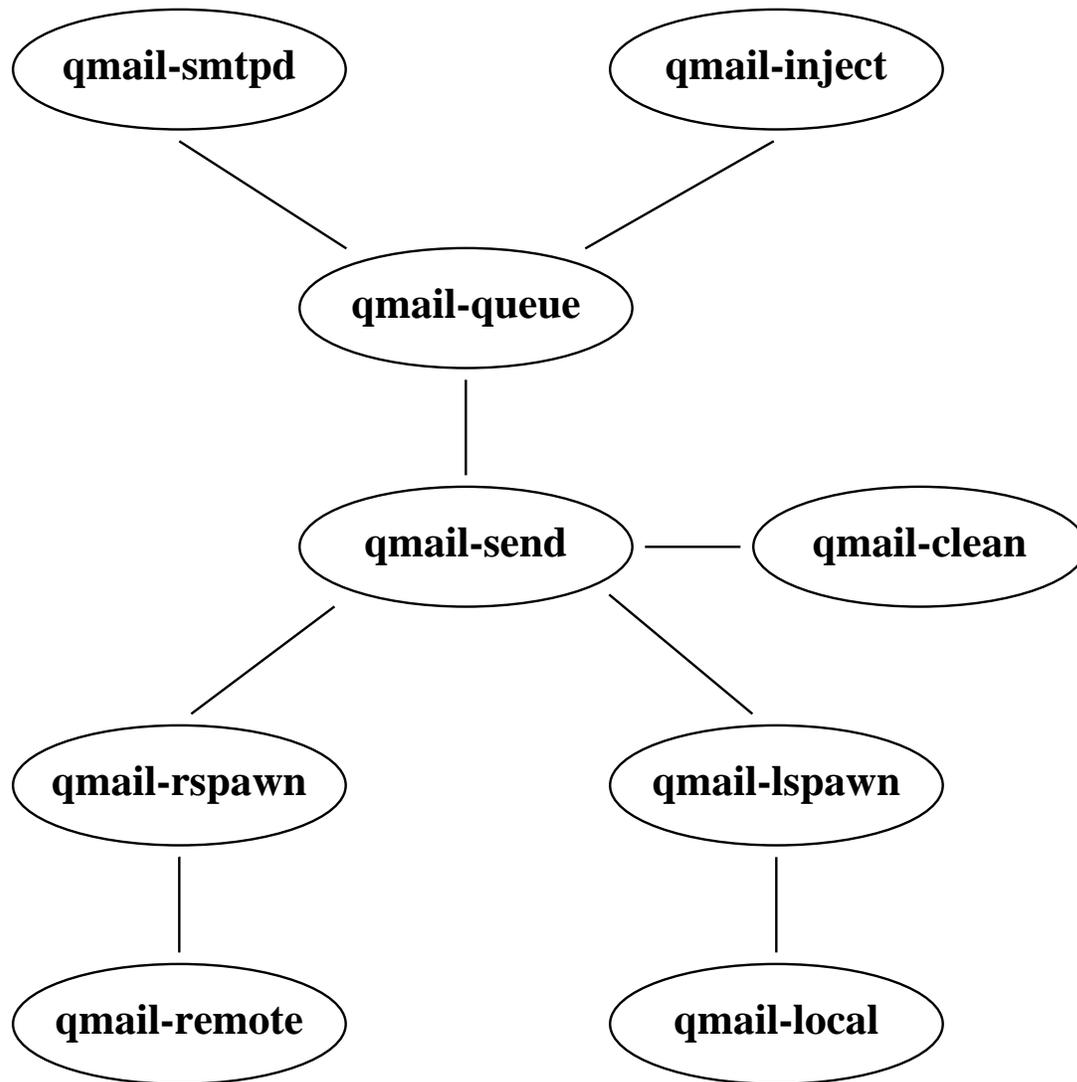
## Disadvantages of using qmail

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.

Qmail cannot be distributed in binary form. Dan wants to make certain that distributions do not unintentionally compromise performance, operation and security. Qmail can be freely distributed in source code form. Dan has plans for releasing binary versions of the, as yet unreleased, 1.02 distribution.

# 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.

## **Availability and competing products**

**qmail** – version 1.01

**<http://www.qmail.org>**

**<ftp://koobera.math.uic.edu/www/qmail.html>**

**sendmail** – version 8.8.8

**<http://www.sendmail.org>**

**exim** – version 1.90

**<http://www.exim.org>**

**smail** – version 3.2

**<ftp://ftp.uu.net/networking/mail/smail>**

**zmailer** – version 2.2e10

**<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.

**qlist** – simple mailing list manager.

**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.

**qsmhook** – enables qmail to use a different MDA. Can be used to maintain sendmail compatibility by delivering to */var/spool/mail/user*. Not recommended.

**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.

**qmsmac** – 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.

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

## **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**

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

Dan is currently working on version 1.02. It still needs a security review and extensive regression testing. He'd also like to see the following features available in subsequent versions of qmail.

**Zero administration for null clients.** Replace SMTP with QMQP for communication between null clients and central mail host.

**Faster installation.** Provide precompiled packages for popular operating systems.

**More forwarding options.** Improvements to qmsmac such as support for separate per-domain forwarding tables, and several front ends supporting different user tastes.

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

## The future

**[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.

**[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.