

Managing Unix-Based Mail Systems



qmail

O'REILLY®

John R. Levine

CHAPTER 8

Delivering and Routing Local Mail

Mail isn't very useful unless it's delivered successfully. This chapter looks at delivering mail addressed to local mailboxes, both for local delivery and for redelivery elsewhere.

Mail to Local Login Users

Local login users usually receive mail in mbox format in `~/Mailbox` and `~/mail.*`.^{*} Or they receive mail in Maildir format in `~/Maildir/`.

Local Delivery .qmail Files and Default Delivery Rules

In the simplest case, a user's `.qmail` file needs to contain only a single line to specify the user's mailbox, either the mbox format mailbox:

```
# deliver into $HOME/Mailbox
./Mailbox
```

or the Maildir:

```
# deliver into a file in $HOME/Maildir/
./Maildir/
```

I suggest that every shell user should have a `.qmail` file (add it to the set of skeleton files that your `adduser` procedure creates), but for users who don't, be sure to set a reasonable default as the argument to `qmail-start` in `/service/qmail/run`, as described in Chapter 3.

^{*} For historical compatibility, some still use `/var/spool/mail/username`, but in this chapter I assume that you have at least moved your users' mailboxes into their home directories where they belong.

Mails and Mail Clients

Although Mails have all sorts of advantages over mboxes, they are not supported in many mail clients. For the popular elm and pine clients, qmail provides small scripts, *elq* and *pinq*, which move mail from the Maildir into an mbox, then run the client. These use the *maildir2mbox* utility, which requires three environment variables to be set. MAIL is the mbox file, usually \$HOME/Mailbox. MAILTMP is the name of a temporary file used to hold a copy of the updated mbox, which must be on the same filesystem as \$MAIL, usually \$HOME/Mailbox.tmp. MAILDIR is the name of the Maildir, usually \$HOME/Maildir.

While these two scripts work adequately, in the long run if you're using Mails, you should use a Maildir client. Unix and Linux command-line users can try mutt, a nice freeware client, Courier IMAP (see Chapter 13), and IMAP clients including pine and the KDE mail client.

Mail Sorting

Unless users receive very little mail, they generally want to sort it before they read it. While Windows mail users tend to pick up all their mail from a single POP mailbox and sort it into local mailboxes in their mail client, Unix users often arrange to sort the mail as it's delivered into mailboxes on the server, and use a client that can handle multiple mailboxes either directly or using IMAP.

There are two general strategies to mail sorting: use multiple incoming addresses or use a filtering program on incoming mail.

Mail Sorting with Subaddresses

The easiest way to sort mailing list mail is to subscribe to each list with a different subaddress. That is, if your address is mary@example.com, you might sign up for three lists as mary-gold@example.com, mary-nade@example.com, and mary-land@example.com.* If your system is set up with per-user subdomains as described in Chapter 12, the three addresses could be written as gold@mary.example.com, nade@mary.example.com, and land@mary.example.com. Then create three files *~mary/.qmail-gold*, *~mary/.qmail-nade*, *~mary/.qmail-land*, each with the delivery instructions for the list mail. If you are using a mail client that handles multiple mailboxes, either directly or through the Courier IMAP server (see Chapter 13), deliver each list to its own mailbox.

This scheme works very well when you only receive mail from a list and you can access the signup through a web site. I use a unique address every time I buy

* These are presumably lists about horticulture, cooking, and geography.

something from a web site that wants an email address. That's useful for both mail sorting and reminding me that a dubious looking piece of mail is in fact from a place to whom I gave the address. It doesn't work so well on discussion lists to which you send as well as receive mail, because it's not easy to put the subaddress on outgoing mail, either to set up the subscription or to send messages to the list. (I've occasionally been reduced to running *qmail-inject* and typing mail headers to it.) It's possible to write a wrapper around *qmail's sendmail* program or *qmail-inject* or, if you're using the QMAILQUEUE patch from Chapter 3, write a wrapper around *qmail-queue* that looks up the destination addresses for a user's outgoing mail in a file and adjusts the return address for mail going to lists. As far as I know, though, nobody's done so. The pragmatic approach is to subscribe both a subaddress and your regular address to a list, and set your regular address to NOMAIL or alias the two together if the list management software permits, so incoming mail from the list goes to the subaddress, while you send outgoing mail from your regular address.

Mail Sorting with Filter Programs

For mail that's sent to a user's regular address, *procmail* and *maildrop* provide flexible script-driven mail sorting. They both provide similar sets of features, with the largest difference being one of style. The *procmail* control language is extremely terse with single-letter commands and options, while *maildrop's* language is more reminiscent of the Unix shells or Perl. *Maildrop* includes some extra features to do simple text processing intended mostly for extracting and handling email addresses, and an optional interface to GDBM keyed files. A significant practical difference is that *procmail* reads an entire message into memory, which means it won't work on very large messages that don't fit. *Maildrop* falls back to temporary files so it can handle even the largest messages, slowly.

I use *procmail*, mostly because I've been using it since before *maildrop* was available. The size limit isn't a problem in practice, because I rarely get mail bigger than 10 MB (certainly not mail that I want), and the filtering I do doesn't need the extra features in *maildrop*.

Mail sorting with *procmail*

Procmail works well when run from *.qmail* files. It expects an mbox-style From line at the beginning of the message, so run it via *preline*:

```
| preline /usr/bin/procmail || exit 111
```

This tells *procmail* to read the standard control file *.procmailrc*, preceded by */etc/procmailrc* if it exists. The `exit 111` is optional, but it's there to ensure that a message stays in the queue if *procmail* crashes, giving you a chance to fiddle around and figure out what went wrong and try again. On the other hand, if your *procmail* script sets the `EXITCODE` variable to return a particular value, you should leave off the `exit` so *qmail* sees your code.

The procmail documentation discusses special provisions for using procmail as a mail delivery agent, and the fine points of its set-uid code. None of this applies to qmail. When procmail starts, whether it's run explicitly from a *.qmail* file or implicitly as the default argument to *qmail-start*, it is like all delivery agents run under the recipient user's ID and home directory. You should *not* install procmail as setuid, because you don't need it and it would be a potential security hole.

If you have multiple mailboxes, either mboxes or Maildir subfolders, procmail can deliver to them directly:

```
# catch messages that appear to be duplicates based on msgid
# (this cryptic recipe cribbed from the procmail examples)
:o Whc: msgid.lock
| formail -D 8192 msgid.cache

# file them in a subfolder
:o a
Maildir/.duplicates/

# deliver mail about breakfast to an mbox
:o
* Subject:.*breakfast
Mail/breakfast

# deliver mail from the lunch list to a Maildir subfolder
# use the List-ID: tag to identify it
:o
* List-ID:.*lunchlist.example.com
Maildir/.lunchlist/

# bounce mail about dinner, we're on a diet
:o
* Subject:.*dinner
{
  EXITCODE=100
}

# deliver everything else to my regular Maildir
:o
Maildir/
```

Note that the Maildir deliveries end with a slash to identify them as Maildirs rather than mboxes, just like in *.qmail* files.

It's quite possible and often useful to combine tagged addresses with procmail. You'll generally want to create separate procmail files for the subaddresses, so put this in *.qmail-color* and tell it to use *procmail-color*:

```
| preline /usr/bin/procmail procmail-color
```

You can use all of qmail's environment variables in your procmail scripts, because procmail makes them available as variables in the script and in the environment of any commands it runs. All but one, that is, because procmail has its own (not very useful) definition of `$DEFAULT`, which overrides qmail's. Fortunately, this is easily circumvented by giving it a different name. Put this in `.qmail-color-default` to refer to the default part of the address as `$$SUBADDR`:

```
| preline /usr/bin/procmail procmail-color SUBADDR="$DEFAULT"
```

Mail sorting with maildrop

In principle, anything you can do with procmail, you can do with maildrop, just differently. In practice, I've found maildrop's code to have severe portability bugs on non-Linux systems, so I can't recommend it for production use, at least not on BSD systems.

To use maildrop, run it from your `.qmail` file:

```
| maildrop
```

The script comes from `/etc/mailfilter` if it exists, then `.mailfilter` in the user's home directory. With no arguments, it delivers mail to the default place determined when maildrop was built, usually `~/Maildir`. Here's the maildrop equivalent of the delivery script:

```
# catch messages that appear to be duplicates based on msgid
# (this code from the maildrop manual)
`reformail -D 8192 duplicate.cache`
if($RETURNCODE == 0)
  to Maildir/.duplicates

# deliver mail about breakfast to an mbox
if(/Subject:.*breakfast/)
  to Mail/breakfast

# deliver mail from the lunch list to a Maildir subfolder
# use the List-ID: tag to identify it
if(/List-ID:.*lunchlist.example.com/)
  to Maildir/.lunchlist/

# bounce mail about dinner, we're on a diet
if(/Subject:.*dinner/)
{
  EXITCODE=100
  exit
}

# deliver everything else to my regular Maildir
to Maildir
```

More Mail Sorting

Although procmail and maildrop are the most popular programs for mail sorting, it's not hard to roll your own. For example, using qmail's *condredirect* program, you can sort mail based on text strings:

```
| condredirect fred-breakfast grep -q 'Subject:.*breakfast'  
| condredirect fred-lunch grep -q 'List-ID:.*lunchlist.example.com'  
Maildir/
```

Because qmail doesn't include separate programs to store mail into mailboxes, conditional deliveries need to use separate subaddresses for each mailbox they use. If you have programs handy to do deliveries (mine's called *mds* for Maildir Store, available at www.qmail.org), you can write these as short shell commands:

```
| if grep -q 'Subject:.*breakfast'; then mds Maildir/.breakfast; exit 99; else exit  
0; esac  
... and so forth ...
```

(The `grep` command reads through the input message, so any program like *mds* has to be sure to rewind its input so it starts delivering the message from the beginning.) For most purposes, you're better off with procmail or maildrop, but if you find you want to do some sorting that you can't easily express in procmail-ese, you can always roll your own.