The Fetchmail FAQ

Table of Contents

Frequently Asked Questions About Fetchmail i

General problems iii
- G1. What is fetchmail and why should I bother? iii
- G2. Where do I find the latest FAQ and fetchmail sources? iii
- G3. Something does not work/I think I've found a bug. Will you fix it? iii
- G4. I have this idea for a neat feature. Will you add it? v
- G5. I want to make fetchmail remove kept mail after some days. v
- G6. Is there a mailing list for exchanging tips? v
- G7. So, what's this I hear about a fetchmail paper? v
- G8. What is the best server to use with fetchmail? vi
- G9. What is the best mail program to use with fetchmail? vi
- G10. How can I avoid sending my password en clair? vi
- G11. Is any special configuration needed to use a dynamic IP address? vii
- G12. Is any special configuration needed to use firewalls? viii
- G13. Is any special configuration needed to send mail? viii
- G14. Is fetchmail Y2K-compliant? viii
- G15. Is there a way in fetchmail to support disconnected IMAP mode? viii
- G16. How will fetchmail perform under heavy loads? viii

Build-time problems xi
- B1. Make coughs and dies when building on FreeBSD. xi
- B2. Lex bombs out while building the fetchmail lexer. xi
- B3. I get link failures when I try to build fetchmail. xi
- B4. I get build failures in the intl directory. xi

Fetchmail configuration file grammar questions xiii
- F1. Why does my old .fetchmailrc file no longer work? xiii
- F2. The .fetchmailrc parser won't accept my all-numeric user name. xiv
- F3. The .fetchmailrc parser won't accept my host or username beginning with 'no'. xiv
- F4. I'm getting a 'parse error' message I don't understand. xv

Configuration questions xvii
- C1. Why do I need a .fetchmailrc when running as root on my own machine? xvii
- C2. How can I arrange for a fetchmail daemon to get killed when I log out? xvii
- C3. How do I know what interface and address to use with --interface? xviii
- C4. How can I set up support for sendmail's anti-spam features? xviii
- C5. How can I poll some of my mailboxes more/less often than others? xix
- C6. Fetchmail works OK started up manually, but not from an init script. xix
- C7. How can I forward mail to another host? xix
- C8. Why is "NOMAIL" an error?/I frequently get messages from cron! xix

How to make fetchmail play nice with various MTAs xxi
- T1. How can I use fetchmail with sendmail? xxi
- T2. How can I use fetchmail with qmail? xxii
- T3. How can I use fetchmail with exim? xxii
- T4. How can I use fetchmail with smail? xxiii
- T5. How can I use fetchmail with SCO's MMDF? xxiii
- T6. How can I use fetchmail with Lotus Notes? xxiii
- T7. How can I use fetchmail with Courier IMAP? xxiii
- T8. How can I use fetchmail with vbmailshield? xxiii

Table of Contents

How to make fetchmail work with various servers ..... **xxv**
S1. How can I use fetchmail with qpopper? xxv
S2. How can I use fetchmail with Microsoft Exchange? xxv
S3. How can I use fetchmail with HP OpenMail? xxvi
S4. How can I use fetchmail with Novell GroupWise? xxvi
S5. How can I use fetchmail with InterChange? xxvi
S6. How can I use fetchmail with MailMax? xxvii
S7. How can I use fetchmail with FTGate? xxvii

How to fetchmail work with specific ISPs ..... **xxix**
I1. How can I use fetchmail with CompuServe RPA? xxix
I2. How can I use fetchmail with Demon Internet's SDPS? xxix
I3. How can I use fetchmail with usa.net's servers? xxx
I4. How can I use fetchmail with geocities POP3 servers? xxx
I5. How can I use fetchmail with Hotmail or Lycos Webmail? xxx
I6. How can I use fetchmail with MSN? xxxi
I7. How can I use fetchmail with SpryNet? xxxi
I8. How can I use fetchmail with comcast.net or other Maillennium servers? xxxi
I9. How can I use fetchmail with GMail/Google Mail? and xxxi
I10. How can I use fetchmail with OAUTH2? xxxi

How to set up well-known security and authentication methods ..... **xxxiii**
K1. How can I use fetchmail with SOCKS? xxxiii
K2. How can I use fetchmail with IPv6 and IPsec? xxxiii
K3. How can I get fetchmail to work with ssh? xxxiii
K4. What do I have to do to use the IMAP-GSS protocol? xxxiii
K5. How can I use fetchmail with SSL or TLS? xxxiv
K6. How can I tell fetchmail not to use TLS if the server advertises it? Why does fetchmail use STARTTLS, STLS, TLS or SSL even though not configured? xxxv

Runtime fatal errors ..... **xxxvii**
R1. Fetchmail isn't working, and -v shows 'SMTP connect failed' messages. xxxvii
R2. When I try to configure an MDA, fetchmail doesn't work. xxxviii
R3. Fetchmail dumps core when given an invalid rc file. xxxviii
R4. Fetchmail dumps core in -V mode, but operates normally otherwise. xxxviii
R5. Running fetchmail in daemon mode doesn't work. xxxviii
R6. Fetchmail randomly dies with socket errors. xxxviii
R7. Fetchmail running as root stopped working after an OS upgrade xxxix
R8. Fetchmail is timing out after fetching certain messages but before deleting them xxxix
R9. Fetchmail is timing out during message fetches xxxix
R10. Fetchmail is dying with SIGPIPE. xxxix
R11. My server is hanging or emitting errors on CAPA. xxxix
R12. Fetchmail isn't working and reports getaddrinfo errors. xxxix
R13. What does "Interrupted system call" mean? xl
R14. Since upgrading fetchmail/OpenSSL, I can no longer connect! xl
R15. Help, I'm getting Authorization failure! xl

Hangs and lockups ..... **xli**
H1. Fetchmail hangs when used with pppd. xli
H2. Fetchmail hangs during the MAIL FROM exchange. xli
H3. Fetchmail hangs while fetching mail. xli

Table of Contents

Disappearing mail ..... **xliii**
D1. I think I've set up fetchmail correctly, but I'm not getting any mail. xliii
D2. All my mail seems to disappear after a dropped connection. xliii
D3. Mail that was being fetched when I interrupted my fetchmail seems to have been vanished. xliii

Multidrop-mode problems ..... **xlv**
M0. What about multidrop, anyways? xlv
M1. I've declared local names, but all my multidrop mail is going to root anyway. xlv
M2. I can't seem to get fetchmail to route to a local domain properly. xlv
M3. I tried to run a mailing list using multidrop, and I have a mail loop! xlvi
M4. My multidrop fetchmail seems to be having DNS problems. xlvi
M5. I'm seeing long DNS delays before each message is processed. xlvi
M6. How do I get multidrop mode to work with majordomo? xlvi
M7. Multidrop mode isn't parsing envelope addresses from my Received headers as it should. xlvii
M8. Users are getting multiple copies of messages. xlvii

Mangled mail ..... **xlix**
X1. Spurious blank lines are appearing in the headers of fetched mail. xlix
X2. My mail client can't see a Subject line. xlix
X3. Messages containing "From" at the start of line are being split. xlix
X4. My mail is being mangled in a new and different way xlix
X5. Using POP3, retrievals seems to be fetching too much! l
X6. My mail attachments are being dropped or mangled. l
X7. Some mail attachments are hanging fetchmail. lii
X8. A spurious) is being appended to my messages. lii
X9. Missing "Content-Transfer-Encoding" header with Domino IMAP. liii
X10. Fetchmail delivers partial messages liii

Other problems ..... **lv**
O1. The --logfile option doesn't work if the logfile doesn't exist. lv
O2. Every time I get a POP or IMAP message, the header is dumped to all my terminal sessions. lv
O3. Does fetchmail reread its rc file every poll cycle? lv
O4. Why do deleted messages show up again when I take a line hit while downloading? lv
O5. Why is fetched mail being logged with my name, not the real From address? lvi
O6. I'm seeing long sendmail delays or hangs near the start of each poll cycle. lvi
O7. Why doesn't fetchmail deliver mail in date-sorted order? lvi
O8. I'm using pppd. Why isn't my monitor option working? lvi
O9. Why does fetchmail keep retrieving the same messages over and over? lvii
O10. Why is the received date on all my messages the same? lvii
O11. I keep getting messages that say "Repoll immediately" in my logs. lvii
O12. Fetchmail no longer expunges mail on a 451 SMTP response. lvii
O13. I want timestamp information in my fetchmail logs. lvii
O14. Fetchmail no longer deletes oversized mails with --flush. lvii
O15. Fetchmail always retains the first message in the mailbox. lviii
O16. Why is the Fetchmail FAQ only available in ISO-216 A4 format? How do I get the FAQ in Letter format? lviii
O17. Linux logs "TCP(fetchmail:...): Application bug, race in MSG_PEEK." lviii

Frequently Asked Questions About Fetchmail

Support? Bug reports? Please read G3 for what information is required to get your problem solved as quickly as possible.

Note that this FAQ is occasionally updated from the Git repository and speaks in the past tense ("since") about a fetchmail release that is not yet available. Please try a release candidate for that version in case you need the new option.

If you have a question or answer you think ought to be added to this FAQ list, file it to one of the trackers at our SourceForge.net project site or post to one of the fetchmail mailing lists (see below).

General problems

G1. What is fetchmail and why should I bother?

Fetchmail is a one-stop solution to the remote mail retrieval problem for Unix machines, quite useful to anyone with an intermittent or dynamic-IP connection to a remote mailserver, SLIP or PPP dialup, or leased line when SMTP isn't desired. Fetchmail can collect mail using any variant of POP or IMAP and forwards to a the local SMTP (via TCP socket) or LMTP (via TCP or Unix socket) listener or into an MDA program, enabling all the normal forwarding/filtering/aliasing mechanisms that would apply to local mail or mail arriving via a full-time TCP/IP connection.

Fetchmail is not a toy or a coder's learning exercise, but an industrial-strength tool capable of transparently handling every retrieval demand from those of a simple single-user ISP connection up to mail retrieval and rerouting for an entire client domain. Fetchmail is easy to configure, unobtrusive in operation, powerful, feature-rich, and well documented.

Fetchmail is Open Source Software. The openness of the sources enables you to review and customize the code, and contribute your changes.

A former fetchmail maintainer once claimed that Open Source software were the strongest quality assurance, but the current maintainers do not believe that open source alone is a criterion for quality – the remotely exploitable POP3 vulnerability (CVE-2005-2335) lingered undiscovered in fetchmail's code for years, which is a hint that open source code does not audit itself.

Fetchmail is licensed under the GNU General Public License v2. Details, including an exception that allows linking against OpenSSL, are in the COPYING file in the fetchmail distribution.

If you found this FAQ in the distribution, see the README for fetchmail's full feature list.

G2. Where do I find the latest FAQ and fetchmail sources?

The latest HTML FAQ is available alongside the latest fetchmail sources at the fetchmail home page: http://www.fetchmail.info/. We used to have new versions in the iBiblio site, but they have stopped accepting uploads, and the fetchmail versions on iBiblio are outdated.

A text dump of this FAQ is included in the fetchmail distribution. Because it freezes at distribution release time, it may not be completely current.

The fetchmail sources are also available in the Git repositories at https://gitlab.com/fetchmail/fetchmail and https://sourceforge.net/p/fetchmail/git/.

Note the Git branches:

- legacy_63 (discontinued, unsupported old branch for 6.3.x releases)
- legacy_64 (stable 6.4.x releases),
- legacy_6x (near-future 6.5.x development),
- next (mid-term future 7.x.x development)

As of 2020-06-21, there no longer is a *master* branch. next was renamed to old-next, and *master* was renamed to **next**.

G3. Something does not work/I think I've found a bug. Will you fix it?

The first thing you should to is to upgrade to the newest version of fetchmail, and then see if the problem reproduces. So you'll probably save us both time if you upgrade and test with the latest version *before* sending in a bug report.

Bugs will be fixed, provided you include enough diagnostic information for me to go on. Send bugs to fetchmail-users. When sending bugs or asking for help, please **do not make up information except your password** and please **report** the following:

1. Your operating system.
2. Your compiler version, if you built from source; otherwise, the name and origin of the RPM or other binary package you installed.
3. The name and version of the SMTP listener or MDA you are forwarding to.
4. Any command-line options you used.
5. The output of **`env LC_ALL=C fetchmail -V`** called with whatever other command-line options you used.
6. **The output of `env LC_ALL=C fetchmail --nodetach -vvv --nosyslog` with whatever other command-line options you use routinely.**

   It is very important that the transcript include your POP/IMAP server's greeting line, so I can identify it in case of server problems. This transcript will not reveal your passwords, which are specially masked out precisely so transcripts can be passed around.

If you have FTP access to your remote mail account, and you have any suspicion that the bug was triggered by a particular message, please include a copy of the message that triggered the bug.

If your bug is something that used to work but stopped working when you upgraded, then you can help pin the bug down by bisecting, i. e. trying intermediate versions of fetchmail until you identify the revision that broke your feature. The smart way to do this is by binary search on the version sequence. First, try the version halfway between your last good one and the current one. If it works, the failure was introduced in the upper half of the sequence; if it doesn't, the failure was introduced in the lower half. Now bisect that half in the same way. In a very few tries, you should be able to identify the exact adjacent pair of versions between which your bug was introduced. **Please** include session transcripts (as described in the last bullet point above) of **both the working and failing versions.** Often, the source of the problem can instantly identified by looking at the differences in protocol transactions.

It may helpful if you include your .fetchmailrc file, but not necessary unless your symptom seems to involve an error in configuration parsing. If you do send in your .fetchmailrc, mask the passwords first! Otherwise, fetchmail -V – as directed above – will usually suffice.

If fetchmail seems to run and fetch mail, but the headers look mangled (that is, headers are missing or blank lines are inserted in the headers) then read the FAQ items in section X before submitting a bug report. Pay special attention to the item on diagnosing mail mangling. There are lots of ways for other programs in the mail chain to screw up that look like fetchmail's fault, but you may be able to fix these by tweaking your configuration.

If the bug involves a core dump or hang, a gdb stack trace is good to have. (Bear in mind that you can attach gdb to a running but hung process by giving the process ID as a second argument.) You will need to reconfigure with:

```
CFLAGS=-g LDFLAGS=" " ./configure
```

Then rebuild in order to generate a version that can be traced with a debugger such as gdb, dbx or idb.

Best of all is a mail file which, when fetched, will reproduce the bug under the latest (current) version.

Any bug I can reproduce will usually get fixed quite quickly. Bugs I can't reproduce are a crapshoot. If the solution isn't obvious when I first look, it may evade me for a long time (or to put it another way, fetchmail is well enough tested that the easy bugs have long since been found). So if you want your bug fixed rapidly, it is not just sufficient but *necessary* that you give me a way to easily reproduce it.

G4. I have this idea for a neat feature. Will you add it?

If it's reasonable for fetchmail and cannot be solved with reasonable effort outside of fetchmail, perhaps.

You can do spam filtering better with procmail or maildrop on the server side and (if you're the server sysadmin) sendmail.cf domain exclusions. If you really want fetchmail to do it from the client side, use a `preconnect` command to call mailfilter.

You can do other policy things better with the `mda` option and script wrappers around fetchmail. If it's a prime-time-vs.-non-prime-time issue, ask yourself whether a wrapper script called from crontab would do the job.

fetchmail's first job is transport though, and it should do this well. If a feature would cause fetchmail to deteriorate in other respects, the feature will probably not be added.

For reasons fetchmail doesn't have other commonly-requested features (such as password encryption, or multiple concurrent polls from the same instance of fetchmail) see ESR's design notes. Note that this document is partially obsoleted by the updated design notes.

G5. I want to make fetchmail remove kept mail after some days.

The second-most-requested feature for fetchmail, after content-based filtering, is the ability to have it remove messages from a maildrop after N days, typically to be used with the `keep` option. Several messaging programs with graphical user interface support this feature.

This feature is not yet implemented. It may be at a future date, spare time of developers permitting.

For the time being, the contrib/ directory contains some *unsupported* tools that may help, namely mold-remover.py and delete-later.

G6. Is there a mailing list for exchanging tips?

There is a fetchmail-users list <fetchmail-users@lists.sourceforge.net> for bug reports and people who want to discuss configuration issues of fetchmail. Please see G3 above for information you need to report.

Then there is a fetchmail-devel list <fetchmail-devel@lists.sourceforge.net> for people who want to discuss fixes and improvements in fetchmail and help co-develop it.

Finally, there is also an announcements-only list, <fetchmail-announce@lists.sourceforge.net>.

For all lists, see https://sourceforge.net/p/fetchmail/mailman/ for subscription, archive and search links.

G7. So, what's this I hear about a fetchmail paper?

Eric S. Raymond also considered fetchmail development a sociological experiment, an extended test to see if his theory about the critical features of the Linux development model was correct.

He considers the experiment a success. He wrote a paper about it titled The Cathedral and the Bazaar which was first presented at Linux Kongress '97 in Bavaria and very well received there. It was also given at Atlanta Linux Expo, Linux Pro '97 in Warsaw, and the first Perl Conference, at UniForum '98, and was the basis of an invited presentation at Usenix '98. The folks at Netscape told ESR it helped them decide to give away the source for Netscape Communicator.

If you're reading a non-HTML dump of this FAQ, you can find the paper on the Web with a search for that title.

G8. What is the best server to use with fetchmail?

Fetchmail will work with any POP, IMAP, ETRN, or ODMR server that conforms to the relevant standards/RFCs (and even some outright broken ones like Microsoft Exchange and Novell GroupWise). This doesn't mean it works equally well with all, however. POP2 servers, and POP3 servers without UIDL, limit fetchmail's capabilities in various ways described on the manual page.

Most modern Unixes (and effectively all Linux/*BSD systems) come with POP3 support preconfigured (but beware of the horribly broken POP3 server mentioned in D2). An increasing minority also feature IMAP (you can detect IMAP support by using the 'Probe for supported protocols' function in the fetchmailconf utility - unfortunately it does not detect SSL-wrapped variants).

If you have the option, we recommend using or installing an IMAP4rev1 or UIDL-capable POP3 server.

A decent POP3/IMAP server that has recently become popular is Dovecot.

G9. What is the best mail program to use with fetchmail?

Fetchmail will work with all popular mail transport programs. It also doesn't care which user agent you use, and user agents are as a rule almost equally indifferent to how mail is delivered into your system mailbox. So any of the popular Unix mail agents – elm, alpine (a rewrite of pine), nmh (the successor to MH), or mutt – will work fine with fetchmail.

All this having been said, I can't resist putting in a discreet plug for mutt. Mutt's interface is only a little different from that of its now-moribund ancestor elm, but its flexibility and excellent handling of MIME and PGP put it in a class by itself. You won't need its built-in POP3 support, though.

G10. How can I avoid sending my password en clair?

You need to ask whether password encryption alone will really address your security exposure. If you think you might be snooped between server and client, it's better to use end-to-end encryption such as GnuPG (see below) on your whole mail stream so none of it can be read.

Then, you can use SSL or TLS for complete end-to-end encryption if you have a TLS-enabled mailserver.

One of the advantages of fetchmail over conventional SMTP-push delivery is that you may be able to arrange encryption by using ssh(1); see K3.

Note that ssh is not a complete privacy solution either, as your mail could have been snooped in transit to your POP server from wherever it originated. For best security, agree with your correspondents to use a tool such as GnuPG (Gnu Privacy Guard) or PGP (Pretty Good Privacy).

If ssh/sshd isn't available, or you find it too complicated for you to set up, password encryption will at least keep a malicious cracker from deleting your mail, and require him to either tap your connection continuously or crack root on the server in order to read it.

You can deduce what encryptions your mail server has available by looking at the server greeting line (and, for IMAP, the response to a CAPABILITY query). Do a `fetchmail -v` to see these, or telnet direct to the server port (110 for POP3, 143 for IMAP).

Your server may have CRAM-MD5 support built in.

The POP3 facility you are most likely to have available is APOP. This is a POP3 feature supported by many servers (fetchmailconf's autoprobe facility will detect it and tell you if you have it). If you see something in the greeting line that looks like an angle-bracket-enclosed Internet address with a numeric left-hand part, that's an APOP challenge (it will vary each time you log in). For some hosts, you need to register a secret on the host (using

`popauth(8)` or some program like that). Specify the secret as your password in your .fetchmailrc; it will be used to encrypt the current challenge, and the encrypted form will be sent back the the server for verification. Note that APOP is no longer considered secure since March 2007.

Alternatively, you may have Kerberos available. This may require you to set up some magic files in your home directory on your client machine, but means you can omit specifying any password at all.

Fetchmail supports two different Kerberos schemes. One is a POP3 variant called KPOP; consult the documentation of your mail server to see if you have it (one clue is the string "krb-IV" in the greeting line on port 110). The other is an IMAP and POP3 facility described by RFC1731 and RFC1734. You can tell if this one is present by looking for AUTH=KERBEROS_V4 in the CAPABILITY response.

Your POP3 server may have the RFC1938 OTP capability to use one-time passwords. To check this, look for the string "otp-" in the greeting line. If you see it, and your fetchmail was built with OPIE support compiled in (see the distribution INSTALL file), fetchmail will detect it also. When using OTP, you will specify a password but it will not be sent en clair.

G11. Is any special configuration needed to use a dynamic IP address?

Yes. In order to avoid giving indigestion to certain picky MTAs (notably exim), fetchmail always makes the RCPT TO address it feeds the MTA a fully qualified one with a hostname part. Normally it does this by appending @ and "localhost", but when you are using Kerberos or ETRN mode it will append @ and your machine's fully-qualified domain name (FQDN).

Appending the FQDN can create problems when fetchmail is running in daemon mode and outlasts the dynamic IP address assignment your client machine had when it started up.

Since the new IP address (looked up at RCPT TO interpretation time) doesn't match the original, the most benign possible result is that your MTA thinks it's seeing a relaying attempt and refuses. More frequently, fetchmail will try to connect to a nonexistent host address and time out. Worst case, you could end up forwarding your mail to the wrong machine!

Use the `smtpaddress` option to force the appended hostname to one with a (fixed) IP address of 127.0.0.1 in your `/etc/hosts`. (The name 'localhost' will usually work; or you can use the IP address itself.)

Only one fetchmail option interacts directly with your IP address, '`interface`'. This option can be used to set the gateway device and restrict the IP address range fetchmail will use. Such a restriction is sometimes useful for security reasons, especially on multihomed sites. See C3.

I recommend against trying to set up the `interface` option when initially developing your poll configuration – it's never necessary to do this just to get a link working. Get the link working first, observe the actual address range you see on connections, and add an `interface` option (if you need one) later.

You can't use ETRN if you have a dynamic IP address (your ISP changes your IP address occasionally, possibly with every connect). You need to have your own registered domain and a definite IP address registered for that domain. The server needs to be configured to accept mail for your domain but then queue it to forward to your machine. ETRN just tells to server to flush its queue for your domain. Fetchmail doesn't actually get the mail in that case.

You can use On-Demand Mail Relay (ODMR) with a dynamic IP address; that's what it was designed for, and it provides capabilities very similar to ETRN. Unfortunately ODMR servers are still not yet widely deployed, as of 2006.

If you're using a dynamic-IP configuration, one other (non-fetchmail) problem you may run into with outgoing mail is that some sites will bounce your email because the hostname you're giving them isn't real (and doesn't match

what they get doing a reverse DNS on your dynamically-assigned IP address). If this happens, you need to hack your sendmail so it masquerades as your host. Setting

```
DMsmarthost.here
```

in your `sendmail.cf` will work, or you can set

```
MASQUERADE_AS(smarthost.here)
```

in the m4 configuration and do a reconfigure. (In both cases, replace `smarthost.here` with the actual name of your mailhost.) See the sendmail FAQ for more details.

G12. Is any special configuration needed to use firewalls?

No. You can use fetchmail with SOCKS, the standard tool for indirecting TCP/IP through a firewall. You can find out about SOCKS, and download the SOCKS software including server and client code, at the ~~http://www.socks.nec.com/~~ *Link defunct* SOCKS distribution site.

The specific recipe for using fetchmail with a firewall is at K1

G13. Is any special configuration needed to *send* mail?

A user asks: but how do we send mail out to the POP3 server? Do I need to implement another tool or will fetchmail do this too?

Fetchmail only handles the receiving side. The sendmail or other preinstalled MTA on your client machine will handle sending mail automatically; it will ship mail that is submitted while the connection is active, and put mail that is submitted while the connection is inactive into the outgoing queue.

Normally, sendmail is also run periodically (every 15 minutes on most Linux systems) in a mode that tries to ship all the mail in the outgoing queue. If you have set up something like pppd to automatically dial out when your kernel is called to open a TCP/IP connection, this will ensure that the mail gets out.

G14. Is fetchmail Y2K-compliant?

Fetchmail is fully Y2K-compliant.

Fetchmail could theoretically have problems when the 32-bit time_t counters roll over in 2038, but I doubt it. Timestamps aren't used for anything but log entry generation. Anyway, if you aren't running on a 64-bit machine by then, you'll deserve to lose.

G15. Is there a way in fetchmail to support disconnected IMAP mode?

No. Fetchmail is a mail transport agent, best understood as a protocol gateway between POP3/IMAP servers and SMTP. Disconnected operation requires an elaborate interactive client. It's a very different problem.

G16. How will fetchmail perform under heavy loads?

Fetchmail streams message bodies line-by-line; the most core it ever requires per message is enough memory to hold the RFC822 header, and that storage is freed when body processing begins. It is, accordingly, quite economical in its use of memory. It will store the UID or UIDL data in core however, which can become considerable if you are keeping lots of messages on the server.

After startup time, a fetchmail running in daemon mode stats its configuration file once per poll cycle to see whether it has changed and should be rescanned. Other than that, a fetchmail in normal operation doesn't touch the disk at all; that job is left up to the MTA or MDA the fetchmail talks to.

Fetchmail's performance is usually bottlenecked by latency on the POP server or (less often) on the TCP/IP link to the server. This is not a problem readily solved by tuning fetchmail, or even by buying more TCP/IP capacity (which tends to improve bandwidth but not necessarily latency).

Build-time problems

~~B1. Make coughs and dies when building on FreeBSD.~~

As of release 6.3.0, fetchmail's Makefile[.in] should work flawlessly with BSD's portable make used on FreeBSD.

B2. Lex bombs out while building the fetchmail lexer.

fetchmail 6.3.0 and newer ship with the lexer and parser in .c formats, so you do not need to use lex unless you hacked the .l or .y files.

fetchmail's lexer has been developed with GNU flex and uses some of its specialties, so the lexer cannot be compiled with the lex tools shipped by some UNIX vendors (HP, SGI, Sun).

B3. I get link failures when I try to build fetchmail.

If you get errors resembling these:

```
mxget.o(.text+0x35): undefined referenceto '__res_search'
mxget.o(.text+0x99): undefined reference to '__dn_skipname'
mxget.o(.text+0x11c): undefined reference to '__dn_expand'
mxget.o(.text+0x187): undefined reference to '__dn_expand'
make: *** [fetchmail] Error 1
```

then you must add "-lresolv" to the LOADLIBS line in your Makefile once you have installed the 'bind' package.

If you get link errors involving `dcgettext`, like these:

```
rcfile_y.o: In function 'yyparse':
rcfile_y.o(.text+0x3aa): undefined reference to 'dcgettext__'
rcfile_y.o(.text+0x4f2): undefined reference to 'dcgettext__'
rcfile_y.o(.text+0x5ee): undefined reference to 'dcgettext__'
rcfile_y.o: In function 'yyerror':
rcfile_y.o(.text+0xc7c): undefined reference to 'dcgettext__'
rcfile_y.o(.text+0xcc8): undefined reference to 'dcgettext__'
rcfile_y.o(.text+0xdf9): more undefined references to 'dcgettext__' follow
```

install an up to date version of GNU gettext, reconfigure and rebuild fetchmail. If that does not help, reconfigure with '--disable-nls' added to the "./configure" command and rebuild.

B4. I get build failures in the intl directory.

Reconfigure with `--disable-nls` and recompile.

Fetchmail configuration file grammar questions

F1. Why does my old .fetchmailrc file no longer work?

If your file predates 6.4.0

Note that fetchmail no longer supports SSLv2, and you should avoid SSLv3 or TLSv1.0 if possible.

If your file predates 6.3.0

The `netsec` option was discontinued and needs to be removed.

If your file predates 5.8.9

If you were using ETRN mode, change your `smtphost` option to a `fetchdomains` option.

If your file predates 5.8.3

The `'via localhost'` special case for use with ssh tunnelling is gone. Use the `%h` feature of `plugin` instead.

If your file predates 5.6.8

In 5.6.8, the `preauth` keyword and option were changed back to `auth`. The `preauth` synonym will still be supported through a few more point releases.

If your file predates 5.6.5

The `imap-gss`, `imap-k4`, and `imap-login` protocol types are gone. This is a result of a major re-factoring of the authentication machinery; fetchmail can now use Kerberos V4 and GSSAPI not just with IMAP but with POP3 servers that have RFC1734 support for the AUTH command.

When trying to identify you to an IMAP or POP mailserver, fetchmail now first tries methods that don't require a password (GSSAPI, KERBEROS_IV); then it looks for methods that mask your password (CRAM-MD5, X-OTP); and only if it the server doesn't support any of those will it ship your password en clair.

Setting the `preauth` option to any value other than 'password' will prevent from looking for a password in your `.netrc` file or querying for it at startup time.

If your file predates 5.1.0

In 5.1.0, the `auth` keyword and option were changed to `preauth`.

If your file predates 4.5.5

If the `dns` option is on (the default), you may need to make sure that any hostname you specify (for mail hosts or for an SMTP target) is a canonical fully-qualified hostname). In order to avoid DNS overhead and complications, fetchmail no longer tries to derive the fetchmail client machine's canonical DNS name at startup.

If your file predates 4.0.6:

Just after the '`via`' option was introduced, I realized that the interactions between the '`via`', '`aka`', and '`localdomains`' options were out of control. Their behavior had become complex and confusing, so much so that I was no longer sure I understood it myself. Users were being unpleasantly surprised.

Rather than add more options or crock the code, I re-thought it. The redesign simplified the code and made the options more orthogonal, but may have broken some complex multidrop configurations.

Any multidrop configurations that depended on the name just after the '`poll`' or '`skip`' keyword being still interpreted as a DNS name for address-matching purposes, even in the presence of a '`via`' option, will break.

It is theoretically possible that other unusual configurations (such as those using a non-FQDN poll name to generate Kerberos IV tickets) might also break; the old behavior was sufficiently murky that we can't be sure. If you think this has happened to you, contact the maintainer.

If your file predates 3.9.5:

The '`remote`' keyword has been changed to '`folder`'. If you try to use the old keyword, the parser will utter a warning.

If your file predates 3.9:

It could be because you're using a .fetchmailrc that's written in the old popclient syntax without an explicit '`username`' keyword leading the first user entry attached to a server entry.

This error can be triggered by having a user option such as '`keep`' or '`fetchall`' before the first explicit username. For example, if you write

```
poll openmail protocol pop3
    keep user "Hal DeVore" there is hdevore here
```

the '`keep`' option will generate an entire user entry with the default username (the name of fetchmail's invoking user).

The popclient compatibility syntax was removed in 4.0. It complicated the configuration file grammar and confused users.

If your file predates 2.8:

The '`interface`', '`monitor`' and '`batchlimit`' options changed after 2.8.

They used to be global options with '`set`' syntax like the batchlimit and logfile options. Now they're per-server options, like '`protocol`'.

If you had something like

```
set interface = "sl0/10.0.2.15"
```

in your .fetchmailrc file, simply delete that line and insert 'interface sl0/10.0.2.15' in the server options part of your 'defaults' declaration.

Do similarly for any '`monitor`' or '`batchlimit`' options.

~~F2. The .fetchmailrc parser won't accept my all-numeric user name.~~

This referred to an older fetchmail 5.x version. Upgrade.

~~F3. The .fetchmailrc parser won't accept my host or username beginning with 'no'.~~

This referred to an older fetchmail 5.x version. Upgrade.

F4. I'm getting a 'parse error' message I don't understand.

The most common cause of mysterious parse errors is putting a server option after a user option. Check the manual page; you'll probably find that by moving one or more options closer to the 'poll' keyword you can eliminate the problem.

Yes, I know these ordering restrictions are hard to understand. Unfortunately, they're necessary in order to allow the 'defaults' feature to work.

Configuration questions

C1. Why do I need a .fetchmailrc when running as root on my own machine?

Ian T. Zimmerman <itz@rahul.net> asked:

On the machine where I'm the only real user, I run fetchmail as root from a cron job, like this:

```
fetchmail -u "itz" -p POP3 -s bolero.rahul.net
```

This used to work as is (with no .fetchmailrc file in root's home directory) with the last version I had (1.7 or 1.8, I don't remember). But with 2.0, it RECPs all mail to the local root user, unless I create a .fetchmailrc in root's home directory containing:

```
skip bolero.rahul.net proto POP3
     user itz is itz
```

It won't work if the second line is just "`user itz`". This is silly.

It seems fetchmail decides to RECP the 'default local user' (i.e. the uid running fetchmail) unless there are local aliases, and the 'default' aliases (itz->itz) don't count. They should.

Answer:

No they shouldn't. I thought about this for a while, and I don't much like the conclusion I reached, but it's unavoidable. The problem is that fetchmail has no way to know, in general, that a local user 'itz' actually exists.

"Ah!" you say, "Why doesn't it check the password file to see if the remote name matches a local one?" Well, there are two reasons.

One: it's not always possible. Suppose you have an SMTP host declared that's not the machine fetchmail is running on? You lose.

Two: How do you know server itz and SMTP-host itz are the same person? They might not be, and fetchmail shouldn't assume they are unless local-itz can explicitly produce credentials to prove it (that is, the server-itz password in local-itz's .fetchmailrc file.).

Once you start running down possible failure modes and thinking about ways to tinker with the mapping rules, you'll quickly find that all the alternatives to the present default are worse or unacceptably more complicated or both.

C2. How can I arrange for a fetchmail daemon to get killed when I log out?

The easiest way to dispatch fetchmail on logout (which will work reliably only if you have just one login going at any time) is to arrange for the command 'fetchmail -q' to be called on logout. Under bash, you can arrange this by putting 'fetchmail -q' in the file '~/.bash_logout'. Most csh variants execute '~/.logout' on logout. For other shells, consult your shell manual page.

Automatic startup/shutdown of fetchmail is a little harder to arrange if you may have multiple login sessions going. In the contrib subdirectory of the fetchmail distribution there is some shell code you can add to your .bash_login and .bash_logout profiles that will accomplish this. Thank James Laferriere <babydr@nwrain.net> for it.

Some people start up and shut down fetchmail using the ppp-up and ppp-down scripts of pppd.

C3. How do I know what interface and address to use with --interface?

This depends a lot on your local networking configuration (and right now you can't use it at all except under Linux and the newer BSDs). However, here are some important rules of thumb that can help. If they don't work, ask your local sysop or your Internet provider.

First, you may not need to use --interface at all. If your machine only ever does SLIP or PPP to one provider, it's almost certainly by a point to point modem connection to your provider's local subnet that's pretty secure against snooping (unless someone can tap your phone or the provider's local subnet!). Under these circumstances, specifying an interface address is fairly pointless.

What the option is really for is sites that use more than one provider. Under these circumstances, typically one of your provider IP addresses is your mailserver (reachable fairly securely via the modem and provider's subnet) but the others might ship your packets (including your password) over unknown portions of the general Internet that could be vulnerable to snooping. What you'll use --interface for is to make sure your password only goes over the one secure link.

To determine the device:

1. If you're using a SLIP link, the correct device is probably sl0.
2. If you're using a PPP link, the correct device is probably ppp0.
3. If you're using a direct connection over a local network such as an ethernet, use the command 'netstat -r' to look at your routing table. Try to match your mailserver name to a destination entry; if you don't see it in the first column, use the 'default' entry. The device name will be in the rightmost column.

To determine the address and netmask:

1. If you're talking to slirp, the correct address is probably 10.0.2.15, with no netmask specified. (It's possible to configure slirp to present other addresses, but that's the default.)
2. If you have a static IP address, run 'ifconfig <device>', where <device> is whichever one you've determined. Use the IP address given after "inet addr:". That is the IP address for your end of the link, and is what you need. You won't need to specify a netmask.
3. If you have a dynamic IP address, your connection IP will vary randomly over some given range (that is, some number of the least significant bits change from connection to connection). You need to declare an address with the variable bits zero and a complementary netmask that sets the range.

To illustrate the rule for dynamic IP addresses, let's suppose you're hooked up via SLIP and your IP provider tells you that the dynamic address pool is 255 addresses ranging from 205.164.136.1 to 205.164.136.255. Then

```
interface "sl0/205.164.136.0/255.255.255.0"
```

would work. To range over any value of the last two octets (65536 addresses) you would use

```
interface "sl0/205.164.0.0/255.255.0.0"
```

C4. How can I set up support for sendmail's anti-spam features?

This answer covers versions of sendmail from 8.9.3-20 (the version installed in Red Hat 6.2) upwards. If you have an older version, upgrade to sendmail 8.9.

Stock sendmails can now do anti-spam exclusions based on a database of filter rules. The human-readable form of the database is at `/etc/mail/access`. The database itself is at `/etc/mail/access.db`.

The table itself uses email addresses, domain names, and network numbers as keys. For example,

```
spammer@aol.com          REJECT
```

```
cyberspammer.com        REJECT
192.168.212             REJECT
```

would refuse mail from spammer@aol.com, any user from cyberspammer.com (or any host within the cyberspammer.com domain), and any host on the 192.168.212.* network. (This feature can be used to do other things as well; see the sendmail documentation for details)

To actually set up the database, run

```
makemap hash deny <deny
```

in /etc/mail.

To test, send a message to your mailing address from that host and then pop off the message with fetchmail, using the -v argument. You can monitor the SMTP transaction, and when the FROM address is parsed, if sendmail sees that it is an address in spamlist, fetchmail will flush and delete it.

Under no circumstances put your **mailhost** or **any host you accept mail from** using fetchmail into your reject file. You **will** lose mail if you do this!!!

C5. How can I poll some of my mailboxes more/less often than others?

Use the *interval* keyword on the ones that should be checked less often. For example, if you do a poll every 5 minutes, and want to poll some mailboxes every 5 minutes and some every 30 minutes, use something like this:

```
poll mainsite.example.com  proto pop3 user ....
poll secondary.example.com proto pop3 interval 6 user ...
```

Then secondary.example.com will be polled every 6th time that mainsite.example.com is polled, which with a polling interval of every 5 minutes means that secondary.example.com will be polled every 30 minutes.

C6. Fetchmail works OK started up manually, but not from an init script.

Often, startup scripts have a different environment than an interactive login shell. For instance, $HOME might point to "/root" when you are logged in as root, but it might be either unset, or set to "/" when the startup scripts are running. That means fetchmail at startup can't find the .fetchmailrc.

Pick a location (such as /etc/fetchmailrc) and use fetchmail's -f option to point fetchmail at it. That should solve the problem.

C7. How can I forward mail to another host?

To forward mail to a host other than the one you are running fetchmail on, use the `smtphost` or `smtpname` option. See the manual page for details.

C8. Why is "NOMAIL" an error?/I frequently get messages from cron!

Some users want to write scripts that take action only if mail could/could not be retrieved, thus fetchmail reports if it has retrieved messages or not.

If you do not want "no mail" to be an error condition (for instance, for cron jobs), use a POSIX-compliant shell and add this to the end of the fetchmail command line, it will change an exit code of 1 to 0 and others to 1:

```
|| [ $? -eq 1 ]
```

If you want to map more than one code to 0, you cannot cascade multiple **|| [$? -eq N]**, but you must instead use the **-o** operator inside the brackets, (see the test(1) manpage for details), such as:

```
|| [ $? -eq 1 -o $? -eq 9 ]
```

A full cron line might then look like this:

```
*/15 * * * * fetchmail -s || [ $? -eq 1 ]
```

How to make fetchmail play nice with various MTAs

T1. How can I use fetchmail with sendmail?

For most sendmails, no special configuration is required. Eric Allman tells me that if `FEATURE(always_add_domain)` is included in sendmail's configuration, you can leave the `rewrite` option off.

If your sendmail complains "sendmail does not relay", make sure your sendmail.cf file says `Cwlocalhost` so that sendmail recognizes 'localhost' as a name of its host.

If you're mailing from another machine on your local network, also ensure that its IP address is listed in ip_allow or name in name_allow (usually in /etc/mail/)

If you find that your sendmail doesn't like the address 'FETCHMAIL-DAEMON@localhost' (which is used in the bouncemail that fetchmail generates), you may have to set `FEATURE(accept_unqualified_senders)`.

Günther Leber reports that Digital Unix sendmails won't work with fetchmail. The symptom is an error message "`553 Local configuration error, hostname not recognized as local`". The problem is that fetchmail normally feeds sendmail with the client machine's host address in the MAIL FROM line. These sendmails think this means they're seeing the result of a mail loop and suppress the mail. You may be able to work around this by running in `--invisible` mode.

If you want to support multidrop mode, and you can get access to your mailserver's sendmail.cf file, it's a good idea to add this rule:

```
H?l?Delivered-To: $h
```

This will cause the mailserver's sendmail to reliably write the appropriate envelope address into each message before fetchmail sees it, and tell fetchmail which header it is. With this change, multidrop mode should work reliably even when the Received header omits the envelope address (which will typically be the case when the message has multiple recipients). However it will still not distinguish the recipients, your only advantage is that no bounce will be sent if a message is BCC addressed to multiple users at your site. To fix even that problem, you might want to try the following hack, which is however untested and quite experimental:

```
H?J?Delivered-To: $u

Mmdrop, P=/usr/bin/procmail, F=lsDFMqSPfhnu9J,
    S=EnvFromSMTP/HdrFromSMTP, R=EnvToSMTP/HdrToSMTP,
    T=DNS/RFC822/X-Unix,
    A=procmail -Y -a $u -d $h
```

For both hacks, you have to declare '`envelope "Delivered-To:"`' on the fetchmail side, to put the virtual domain (e.g. 'domain.com') with RELAY permission into your access file and to add a line reading '`domain.com local:local-pop-user`' for the first and '`domain.com mdrop:local-pop-user`' for the second hack to your mailertable.

You will notice that if the mail already has a Delivered-To header, sendmail will not add another. Further, editing sendmail.cf directly is not very comfortable. Solutions for both problems can be found in Peter 'Rattacresh' Backes' 'hybrid' patch against sendmail. Have a look at it, you can find it in the contrib subdirectory.

Feel free to try Martijn Lievaart's detailed recipe in the contrib subdirectory of the fetchmail source distribution, it attempts to realize multidrop mailboxes with an external script.

If for some reason you are invoking sendmail via the `mda` option (rather than delivering to port 25 via smtp), don't forget to include the -i switch. Otherwise you will occasionally get mysterious delivery failures with a SIGPIPE as the sendmail instance dies. The problem is messages with a single dot at start of a text line.

T2. How can I use fetchmail with qmail?

qmail as your local SMTP server

Don't! Avoid qmail and netqmail, they are broken and unmaintained.

Turn on the `forcecr` option; qmail's listener mode doesn't like header or message lines terminated with bare linefeeds.
(This information contributed by Robert de Bath <robert@mayday.cix.co.uk>.)

qmail as your ISP's POP3 server

Note that qmail's POP3 server, as of version 1.03 and netqmail 1.05, miscalculates the message sizes, so you may see size-related fetchmail warnings.

If a mailhost is using the qmail package, then it is usually possible to set up one fetchmail link to reliably collect the mail for an entire domain.

One of the basic features of qmail is the 'Delivered-To:' message header. Whenever qmail delivers a message to a local mailbox it puts the username and hostname of the envelope recipient on this line. One major reason for this is to prevent mail loops, the other is to transport envelope information which is essential for multidrop (domain-in-a-mailbox) schemes.

To set up qmail to batch mail for a disconnected site, the ISP-mailhost will have normally put that site in its 'virtualhosts' control file so it will add a prefix to all mail addresses for this site. This results in mail sent to 'username@userhost.userdom.example.com' having a 'Delivered-To:' line of the form:

```
Delivered-To: mbox-userstr-username@userhost.userdom.example.com
```

A single host maildrop will be slightly simpler:

```
Delivered-To: mbox-userstr-username@userhost.example.com
```

The ISP can make the 'mbox-userstr-' prefix anything they choose but a string matching the user host name is likely.

To use this line you must:

1. Ensure the option '`envelope "Delivered-To"`' is in the fetchmail config file.
2. Ensure the option '`qvirtual "mbox-userstr-"`' is in the fetchmail config file, in order to remove this prefix from the username. (added by Luca Olivetti)
3. Ensure you have a `localdomains` option containing '`userdom.example.com`' or '`userhost.userdom.example.com`' respectively.

T3. How can I use fetchmail with exim?

If you have `rewrite` on:

There is an RFC1123 requirement that MAIL FROM and RCPT TO addresses you pass to it have to be canonical (e.g. with a fully qualified hostname part). Therefore fetchmail tries to pass fully qualified RCPT TO addresses. But exim does not by default accept 'localhost' as a fully qualified domain. This can be fixed.

In exim.conf, add 'localhost' to your local_domains declaration if it's not already present. For example, the author's site at thyrsus.com would have a line reading:

```
local_domains = thyrsus.com:localhost
```

If you have `rewrite` off:

MAIL FROM is a potential problem if the MTAs upstream from your fetchmail don't necessarily pass canonicalized From and Return-Path addresses, and fetchmail's `rewrite` option is off. The specific case where this has come up involves bounce messages generated by sendmail on your mailer host, which have the (un-canonicalized) origin address MAILER-DAEMON.

The right way to fix this is to enable the `rewrite` option and have fetchmail canonicalize From and Return-Path addresses with the mailserver hostname before exim sees them. This option is enabled by default, so it won't be off unless you turned it off.

If you must run with `rewrite` off, there is a switch in exim's configuration files that allows it to accept domainless MAIL FROM addresses; you will have to flip it by putting the line

```
sender_unqualified_hosts = localhost
```

in the main section of the exim configuration file. Note that this will result in such messages having an incorrect domain name attached to their return address (your SMTP listener's hostname rather than that of the remote mail server).

T4. How can I use fetchmail with smail?

Smail 3.2 is very nearly plug-compatible with sendmail, and may work fine out of the box.

We have one report that when processing multiple messages from a single fetchmail session, smail sometimes delivers them in an order other than received-date order. This can be annoying because it scrambles conversational threads. This is not fetchmail's problem, it is an smail 'feature' and has been reported to the maintainers as a bug.

Very recent smail versions require an `-smtp_hello_verify` option in the smail config file. This overrides smail's check to see that the HELO address is actually that of the client machine, which is never going to be the case when fetchmail is in the picture. According to RFC1123 an SMTP listener *must* allow this mismatch, so smail's new behavior (introduced sometime between 3.2.0.90 and 3.2.0.95) is a bug.

You may also need to say `-smtp_hello_broken_allow=127.0.0.1` in order for smail to accept the "localhost" that fetchmail normally appends to recipient addresses.

T5. How can I use fetchmail with SCO's MMDF?

MMDF itself is difficult to configure, but it turns out that connecting fetchmail to MMDF's SMTP channel isn't that hard. You can read an MMDF recipe that describes replacing a UUCP link with fetchmail feeding MMDF.

T6. How can I use fetchmail with Lotus Notes?

The Lotus Notes SMTP gateway tries to deduce when it should convert \n to \r\n, but its rules are not the intuitive and correct-for-RFC822 ones. Use 'forcecr'.

T7. How can I use fetchmail with Courier IMAP?

The courier mta doesn't like RCPT addresses that look like `someone@localhost`. Work around this with an `smtphost` or `smtpaddress`.

T8. How can I use fetchmail with vbmailshield?

vbmailshield's SMTP interpreter is broken. It doesn't understand RSET.

As a workaround, you can set batchlimit to 1 so RSET is never used.

How to make fetchmail work with various servers

~~S1. How can I use fetchmail with qpopper?~~

The information that used to be here was obsolete and dropped.

S2. How can I use fetchmail with Microsoft Exchange?

It's been reliably reported that Exchange 2000's POP3 support is so broken that it's unusable. One symptom is that messages without a terminating newline get the POP3 message termination dot emitted -- you guessed it -- right after the last character of the message, with no terminating newline added. This will hang fetchmail or any other RFC-compliant server. IMAP is alleged to work OK, though.

Older versions of Exchange are semi-usable. They randomly drop attachments on the floor, though. Microsoft acknowledges this as a known bug and apparently has no plans to fix it.

Fetchmail using IMAP usually supports the proprietary NTLM mode used with Microsoft Exchange servers. "Usually" here means that it fails on some servers for reasons that we haven't been able to debug yet, perhaps it's related to the NTLM domain.

To enable this NTLM mode, configure fetchmail with the --enable-NTLM option and recompile it. Specify a user option value that looks like 'user@domain': the part to the left of the @ will be passed as the username and the part to the right as the NTLM domain.

Microsoft Exchange violates the POP3 and IMAP RFCs. Its LIST command does not reveal the real sizes of mail in the pop mailbox, but the sizes of the compressed versions in the exchange mail database (thanks to Arjan De Vet and Guido Van Rooij for alerting us to this problem).

Fetchmail works with Microsoft Exchange, despite this brain damage. Two features are compromised. One is that the --limit option will not work right (it will check against compressed and not actual sizes). The other is that a too-small SIZE argument may be passed to your ESMTP listener, assuming you're using one (this should not be a problem unless the actual size of the message is above the listener's configured length limit).

ESR learned that there's supposed to be a registry bit that can fix this breakage:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\MsExchangeIs\Parameters
System\Pop3 Compatibility
```

This is a bitmask that controls the variations from the standard protocol. The bits defined are:

0x00000001:
: Report exact message sizes for the LIST command

0x00000002:
: Allow arbitrary linear whitespace between commands and arguments

0x00000004:
: Enable the LAST command

0x00000008:
: Allow an empty PASS command (needed for users with blank passwords, but illegal in the protocol)

0x00000010:
: Relax the length restrictions for arguments to commands (protocol requires 40, but some user names may be longer than that).

0x00000020:
: Allow spaces in the argument to the USER command.

There's another one that may be useful to know about:

```
KEY_LOCAL_MACHINE\System\CurrentControlSet\Services\MsExchangeIs\Parameters
System\Pop3 Performance
```

0x00000001:
: Render messages to a temporary stream instead of sending directly from the database (should always be on)

0x00000002: Flag unrenderable messages (instead of just failing commands) (should only be on if you are seeing the problems reported in KB Q168109)

0x00000004:
: Return from the QUIT command before all messages have been deleted.

The Microsoft employee who revealed this information to ESR admitted that he couldn't find it anywhere in their public knowledge base.

Another specific problem we have seen with Exchange servers has as its symptom a response to LOGIN that says "NO Ambiguous Alias". Grant Edwards writes:

> This means that Exchange Server is too [...] stupid to figure out which mailbox belongs to you. Instead of actually keeping track of which inbox belongs to which user, it uses some half-witted, guess-o-matic heuristic to try to guess your mailbox name from your username.
>
> In your case it doesn't work because your username maps to more than one mailbox. For some people it doesn't work because their username maps to zero mailboxes.
>
> You've got several options:
>
> - Get your administrator to configure the server so that usernames and mailbox names are the same.
> - Get your administrator to add an alias that maps your username explicitly to your mailbox name.

But, the best option involves finding a server that runs better software.

S3. How can I use fetchmail with HP OpenMail?

No special configuration is required, but OpenMail versions prior to 6.0 have an annoying bug similar to the big one in Microsoft Exchange. The message sizes it gives in the LIST are rounded to the nearest 1024 bytes. It also has a nasty habit of discarding headers it doesn't recognize, such as X- and Resent- headers.

OpenMail's project manager claims these bugs have been fixed in 6.0.

We've had a more recent report (December 2001) that the TOP command fails, returning only one line regardless of its argument, on something identifying itself as "OpenMail POP3 interface".

S4. How can I use fetchmail with Novell GroupWise?

The Novell GroupWise IMAP server is (according to the designer of IMAP) unusably broken. Among other things, it doesn't include a required content length in its BODY[TEXT] response.

Fetchmail works around this problem to some extent, but no guarantees.

S5. How can I use fetchmail with InterChange?

You can't. At least not if you want to be able to see attachments. InterChange has a bug similar to the MailMax server (see below): it reports the message length with attachments but doesn't download them on TOP or RETR.

On Jan 9 2001, the people at InfiniteMail sent ESR mail informing him that their new 3.61.08 release of InterChange fixed this problem.

S6. How can I use fetchmail with MailMax?

You can't. At least not if you want to be able to see attachments. MailMax has a bug; it reports the message length with attachments but doesn't download them on TOP or RETR.

Also, we're told that TOP sometimes fails to retrieve the entire message even when enough lines have been specified. The MailMax developers have acknowledged this bug as of 4 May 2000, but there is no fix yet. If you must use this server, force RETR with the `fetchall` option.

S7. How can I use fetchmail with FTGate?

The FTGate V2 server (and possibly older versions as well) has a weird bug. It answers OK twice to a TOP request! Use the `fetchall` option to force use of RETR and work around this bug.

How to fetchmail work with specific ISPs

I1. How can I use fetchmail with CompuServe RPA?

First, make sure your fetchmail has the RPA support compiled in. Stock fetchmail binaries (such as you might get from an RPM) don't. You can check this by looking at the output of `fetchmail -V`; if you see the string "+RPA" after the version ID you're good to go, otherwise you'll have to build your own from sources (see the INSTALL file in the source distribution for directions).

Give your CompuServe pass-phrase in lower case as your password. Add '@compuserve.com' to your user ID so that it looks like 'user <UserID>@compuserve.com', where <UserID> can be either your numerical userID or your E-mail nickname. An RPA-enabled fetchmail will automatically check for csi.com in the POP server's greeting line. If that's found, and your user ID ends with '@compuserve.com', it will query the server to see if it is RPA-capable, and if so do an RPA transaction rather than a plain-text password handshake.

Warning: the debug (-v -v) output of fetchmail will show your pass-phrase in Unicode!

These two .fetchmailrc entries show the difference between an RPA and non-RPA configuration:

```
# This version will use RPA
poll csi.com via "pop.site1.csi.com" with proto POP3 and options no dns
    user "CSERVE_USER@compuserve.com" there with password "CSERVE_PASSWORD"
        is LOCAL_USER here options fetchall stripcr

# This version will not use RPA
poll non-rpa.csi.com via "pop.site1.csi.com" with proto POP3 and options no dns
    user "CSERVE_USER" there with password "CSERVE_POP3_PASSWORD"
       is LOCAL_USER here options fetchall stripcr
```

I2. How can I use fetchmail with Demon Internet's SDPS?

Single-drop mode

You can get fetchmail to download the email for just one user from Demon Internet's POP3 server by giving it a username consisting of your Demon user name followed by your account name, with an at-sign between them.

For example, to download email for the user <philh@vision25.demon.co.uk>, you could use the following .fetchmailrc file:

```
set postmaster "philh"
poll pop3.demon.co.uk with protocol POP3:
    user "philh@vision25" is philh
```

Multi-drop mode

Demon Internet's SDPS service is an implementation of POP3. All messages have a Received: header added when they enter the maildrop, like this:

```
Received: from punt-1.mail.demon.net by mailstore for fred@xyz.demon.co.uk
          id 899963657:10:27896:0; Thu, 09 Jul 98 05:54:17 GMT
```

To enable multi-drop mode you need to tell fetchmail that 'mailstore' is the name of the host which accepted the mail, and let it know the hostname part(s) of your E-mail address. The following example assumes that your hostname is xyz.demon.co.uk, and that you have also bought "mail forwarding" for the domain my-company.co.uk (in which case your MTA must also be configured to accept mail sent to user@my-company.co.uk)

```
poll pop3.demon.co.uk proto pop3 aka mailstore no dns:
  localdomains xyz.demon.co.uk my-company.co.uk
```

```
user xyz is *
```

Note that Demon may delete mail on the server which is more than 30 days old; see their POP3 page for details.

The SDPS extension

There's a different way to do multidrop. It's not necessary on Demon Internet, since fetchmail can parse Received addresses, but the person who implemented this didn't know that. It may be useful if Demon Internet ever changes mail transports.

SDPS includes a non-standard extension for retrieving the envelope of a message (*ENV), which fetchmail optionally supports if compiled with the --enable-SDPS option. If you have it, the first line of the fetchmail -V response will include the string "+SDPS".

Once you have SDPS compiled in, fetchmail in POP3 mode will automatically detect when it's talking to a Demon Internet host in multidrop mode, and use the *ENV extension to get an envelope To address.

The autodetection works by looking at the hostname in the POP3 greeting line; if you're accessing Demon Internet through a proxy it may fail. To force SDPS mode, pick "sdps" as your protocol.

I3. How can I use fetchmail with usa.net's servers?

Enable '`fetchall`'. A user reports that the 2.2 version of USA.NET's POP server reports that you must use the '`fetchall`' option to make sure that all of the mail is retrieved, otherwise some may be left on the server. This is almost certainly a server bug.

The usa.net servers (at least in their 2.2 version, June 1998) don't handle the TOP command properly, either. Regardless of the argument you give it, they retrieve only about 10 lines of the message. Fetchmail normally uses TOP for message retrieval in order to avoid marking messages seen, but '`fetchall`' forces it to use RETR instead.

Also, we're told USA.NET adds a ton of hops to your messages. You may need to raise the MaxHopCount parameter in your sendmail.cf to avoid having fetched mail rejected.

I4. How can I use fetchmail with geocities POP3 servers?

Nathan Cutler reports that the the mail.geocities.com POP3 servers fail to include the first Received line of the message in the send to fetchmail. This can solve problems if your MUA interprets Received continuations as body lines and doesn't parse any of the following headers.

Workaround is to use "mda" keyword or "--mda" switch:

```
mda "sed -e '1s/^\t/Received: /' | formail | /usr/bin/procmail -d <user>"
```

Replace \t with exactly one tabulation character.

You should also consider using "fetchall" option because Geocities' servers sometimes think that the first 45 messages have already been read.

~~I5. How can I use fetchmail with Hotmail or Lycos Webmail?~~

It appears that both services offer POP3 or IMAP access these days.

Other than that, it might be possible to use the HotWayDaemon daemon. You don't even need to install hotwayd as a daemon in `inetd.conf` but can use it as a plugin. Your configuration should look like this:

```
poll localhost protocol pop3 tracepolls
   plugin "/usr/local/sbin/hotwayd -l 0 -p yourproxy:yourproxyport"
   username "youremail@hotmail.com" password "yourpassword"
   fetchall
```

As a second option you may consider using GetLive, a successor to the discontinued Gotmail.

I6. How can I use fetchmail with MSN?

See I5 above.

I7. How can I use fetchmail with SpryNet?

SpryNet is no more.

I8. How can I use fetchmail with comcast.net or other Maillennium servers?

Stock fetchmail will work with a Maillennium POP3/PROXY server... *but* this server will truncate "TOP" responses after 64 - 82 kB (we have varying reports), in violation of Internet Standard #53 aka. RFC-1939 (POP3). Don't mistake this for a fetchmail bug. (Reported July 2003.) Comcast documented they haven't understood what this is about in two messages from April 2004.

Beginning with version 6.3.2, fetchmail will fall back to the RETR command if the greeting string contains "Maillennium POP3/PROXY server", and print a warning message. This means however that fetchmail has no means to prevent the "seen" flag from being set on the server (Note that officially, POP3 has no notion of seen tracking, but it works for some sites.)

Workaround for older versions: use the `fetchall` option.

I9. How can I use fetchmail with GMail/Google Mail? and

I10. How can I use fetchmail with OAUTH2?

You cannot. Use something else. So far the maintainer is only aware of abusive OAUTH2 schemes that take end user's freedom of choice away.

Google and Microsoft have started pushing towards more complex authentication schemes based on OAuth 2.0 that require clients and users to jump through quite a few hoops, and use web browsers for signing in, and software vendors to hand in their software for sometimes paid reviews. Such is not going to happen for fetchmail. If this hinders access to your account through fetchmail, you may need to turn on access for "less secure apps", or create an application or service specific password. For Google, this - at some point in time - used to live at https://myaccount.google.com/lesssecureapps.
It is disputable whether an application that does not include web browsing capabilities or heavy-weight libraries is "less secure" as Google claims.

Google's IMAP servers, as of April 2008, are broken and re-encode MIME-encoded headers improperly and are not feature-complete yet. The model how their servers organize mail also deviates in significant ways from what the POP3 or IMAP protocol 'fathers' conceived. This means all sorts of strange effects, for instance, your sent mail may show up in the mail that fetchmail fetches. It's best to avoid fetching mail from Google until they are using standards-compliant software.

If you still need to use Google's mail service, these links may help (valid as of 2011-04-13):

- Other ways to access Gmail > POP

- Other ways to access Gmail > IMAP
- Using POP on multiple clients or mobile devices
- Some [POP3] mail was not downloaded
- I'm having problems downloading [IMAP] mail

How to set up well-known security and authentication methods

K1. How can I use fetchmail with SOCKS?

Giuseppe Guerini added a **`--with-socks`** compile-time option that supports linking with socks library. If you specify the value of this option as "yes", the configure script will try to find the Rconnect library and set the makefile up to link it. You can also specify a directory containing the Rconnect library.

Alan Schmitt has added a similar **`--with-socks5`** option that may work better if you have a recent version of the SOCKS library.

In either case, fetchmail has no direct configuration hooks, but you can specify which socks configuration file the library should read by means of the `SOCKS_CONF` environment variable. In order to bypass the SOCKS proxy altogether, you could run (adding your usual options to the end of this line):

```
env SOCKS_CONF=/dev/null fetchmail
```

K2. How can I use fetchmail with IPv6 and IPsec?

To use fetchmail with IPv6, you need a system that supports IPv6, the "Basic Socket Interface Extensions for IPv6" (RFC 2133).

The NRL IPv6+IPsec software distribution can be obtained from: http://web.mit.edu/network/isakmp/

More information on using IPv6 with Linux can be obtained from:

- http://www.bieringer.de/linux/IPv6/IPv6-HOWTO/IPv6-HOWTO.html

K3. How can I get fetchmail to work with ssh?

Use the `plugin` option. This is dead simple with IMAP:

```
plugin "ssh %h /usr/sbin/imapd"
```

You may have to use a different absolute pathname, whatever the location of imapd on your mailserver is. This option tells fetchmail that instead of opening a connection on the server's port 143 and doing standard IMAP authentication, fetchmail should ssh to the server and run imapd, using the more secure ssh authentication (as well as getting ssh's end-to-end encryption). Most IMAP daemons will detect that they've been called from the command line and assume the connection is preauthenticated.

POP3 daemons aren't quite as smart. They won't know they are preauthenticated in this mode, so you'll actually have to ship your password. It will be under ssh encryption, though, so that shouldn't be a problem.

K4. What do I have to do to use the IMAP-GSS protocol?

Fetchmail can use RFC1731 GSSAPI authorization to safely identify you to your IMAP server, as long as you can share Kerberos V credentials with your mail host and you have a GSSAPI-capable IMAP server.

fetchmail does not compile in support for GSS by default, since it requires libraries from a Kerberos V distribution, such as MIT Kerberos or Heimdal Kerberos.

If you have these, compiling in GSS support is simple: add a `--with-gssapi=[/path/to/krb5/root]` option to configure. For instance, I have all of my Kerberos V libraries installed under /usr/krb5 so I run

```
configure --with-gssapi=/usr/krb5
```

Setting up Kerberos V authentication is beyond the scope of this FAQ (you may find Jim Rome's paper How to Kerberize your site helpful), but you'll at least need to add a credential for imap/[mailhost] to the keytab of the mail server (IMAP doesn't just use the host key). Then you'll need to have your credentials ready on your machine (cf. kinit).

After that things are very simple. Set your protocol to imap-gss in your .fetchmailrc, and omit the password, since imap-gss doesn't need one. You can specify a username if you want, but this is only useful if your mailbox belongs to a username different from your Kerberos principal.

Now you don't have to worry about your password appearing in cleartext in your .fetchmailrc, or across the network.

K5. How can I use fetchmail with SSL or TLS?

You'll need to have the OpenSSL or wolfSSL libraries installed. See the relevant README.SSL file that shipped with your fetchmail for the exact versions needed. Configure with --with-ssl (default since fetchmail v6.4.0). If you have the OpenSSL libraries installed in commonly-used default locations, or the library comes with a pkg-config compatible .pc file on your system, this will often suffice. If you have them installed in a non-default location, you'll need to specify the OpenSSL installation directory as an argument to --with-ssl after an equal sign.

Fetchmail binaries built this way support `ssl` and `tls`, `sslkey`, and `sslcert` options that control SSL encryption, and will automatically try to negotiate `starttls` or `stls`if the server offers it. You will need to have an SSL/TLS-enabled mailserver to use these options. See the manual page for details and some words of care on the limited security provided.

If your open OpenSSL session dies with a message that complains "PRNG not seeded", update or improve your operating system. This means that the OpenSSL library on your machine has been unable to locate a source of random bits from which to seed its random-number generator; normally these come from the `/dev/urandom`, and this message probably means your OS doesn't have that device.

An interactive program could seed the random number generator from keystroke timings or some other form of user input. Because fetchmail is primarily designed to run forever as a background daemon, that option is not available in this case.

If you don't have the libraries installed, but do have the OpenSSL utility toolkit, something like this may work (but will not authenticate the server):

```
poll MYSERVER port 993 plugin "openssl s_client -connect %h:%p"
        protocol imap username MYUSERNAME password MYPASSWORD
```

You should note that SSL or TLS are only secure against a "man-in-the-middle" attack if the client is able to verify that the presented peer's public key is the correct one, and has not been substituted by an attacker along the way. fetchmail can do this in one of two ways: by verifying the SSL certificate, or by checking the fingerprint of the peer's public key.

There are three parts to TLS certificate verification: checking that the domain name in the certificate matches the hostname you asked to connect to; checking that the certificate expiry date has not passed; and checking that the certificate has been signed by a known Certificate Authority (CA). This last step takes some preparation, as you need to install the root certificates of all the CA's which you might come across.

The easiest way to do this is using the root CA keys supplied in the OpenSSL distribution, which means you need to download and unpack the source tarball from www.openssl.org. Once you have done that:

1. `mkdir /etc/ssl/certs`
2. in the openssl-x.x.x/certs directory: `cp *.pem /etc/ssl/certs/`

3. in the openssl-x.x.x/tools directory: edit c_rehash and set `$dir="/etc/ssl"`
4. run "perl c_rehash". This generates a number of symlinks within the /etc/ssl/certs/ directory

Now in .fetchmailrc, set option sslcertpath to point to this directory:

```
poll pop3.example.com proto pop3 uidl no dns
  user foobar@example.com password xyzzy is foobar ssl sslcertpath /etc/ssl/certs
```

If the server certificate has not been signed by a known CA (e.g. it is a self-signed certificate), then this certificate validation will always fail.

Certificate verification is always attempted. If it fails, since v6.4.0, by default the connection aborts (6.3 and older would carry on after printing a warning, unless `--sslcertck` was in effect). If your server doesn't have a valid certificate though (e.g. it has a self-signed certificate) then it will never verify, and the only way you can protect yourself is by checking the fingerprint. You should then contact the operator and as for properly issued certificates.

To check the peer fingerprint: first use fetchmail -v once to connect to the host, at a time when you are pretty sure that there is no attack in progress (e.g. you are not traversing any untrusted network to reach the server). Make a note of the fingerprint shown. Now embed this in your .fetchmailrc using the sslfingerprint option: e.g.

```
poll pop3.example.com proto pop3 uidl no dns
  user foobar@example.com password xyzzy is foobar
  ssl sslfingerprint "67:3E:02:94:D3:5B:C3:16:86:71:37:01:B1:3B:BC:E2"
```

When you next connect, the public key presented by the server will be verified against the fingerprint given. If it's different, it may mean that a man-in-the-middle attack is in progress - or it might just mean that the server changed its key. It's up to you to determine which has happened.

K6. How can I tell fetchmail not to use TLS if the server advertises it? Why does fetchmail use STARTTLS, STLS, TLS or SSL even though not configured?

Some options in fetchmail - including sslcertck that has become the default in v6.4.0 - require fetchmail to negotiate SSL or TLS. In other situations, fetchmail tries to negotiate SSL or TLS opportunistically: Some servers advertise STLS (POP3) or STARTTLS (IMAP), and fetchmail will automatically attempt TLS negotiation if SSL was enabled at compile time. This can however cause problems if the upstream didn't configure his certificates properly.

In some situations, the server does not offer STARTTLS or STLS, but it would offer a TLS-wrapped operation on a dedicated, separate port. In such a situation, adding `ssl` to the rcfile (or `--ssl` on the command line) is all there is to it. Fetchmail will use the default port for the "secure" service and negotiate TLS or SSL right away.

In order to prevent fetchmail 6.4.0 and newer versions from trying STLS or STARTTLS negotiation, and only as a last resort because it exposes all communication to potential eavesdroppers, you could add this option:

```
sslproto ''
```

TLS if the server advertises it? Why does fetchmail use STARTTLS, STLS, TLS or SSL even though not configured?

Runtime fatal errors

R1. Fetchmail isn't working, and -v shows 'SMTP connect failed' messages.

Fetchmail itself is probably working, but your SMTP port 25 listener is down or inaccessible.

The first thing to check is if you can telnet to port 25 on your smtp host (which is normally 'localhost' unless you've specified an smtp option in your .fetchmailrc or on the command line) and get a greeting line from the listener. If the SMTP host is inaccessible or the listener is down, fix that first.

In Red Hat Linux 6.x, SMTP is disabled by default. To fix this, set "DAEMON=yes" in your /etc/sysconfig/sendmail file, then restart sendmail by running "/sbin/service sendmail restart".

If the listener seems to be up when you test with telnet, the most benign and typical problem is that the listener had a momentary seizure due to resource exhaustion while fetchmail was polling it -- process table full or some other problem that stopped the listener process from forking. If your SMTP host is not 'localhost' or something else in /etc/hosts, the fetchmail glitch could also have been caused by transient nameserver failure.

Try running fetchmail -v again; if it succeeds, you had one of these kinds of transient glitch. You can ignore these hiccups, because a future fetchmail run will get the mail through.

If the listener tests up, but you have chronic failures trying to connect to it anyway, your problem is more serious. One way to work around chronic SMTP connect problems is to use --mda. But this only attacks the symptom; you may have a DNS or TCP routing problem. You should really try to figure out what's going on underneath before it bites you some other way.

We have one report (from toby@eskimo.com) that you can sometimes solve such problems by doing an `smtp` declaration with an IP address that your routing table maps to something other than the loopback device (he used ppp0).

We also have a report that this error can be caused by having an /etc/hosts file that associates your client host name with more than one IP address.

It's also possible that your DNS configuration isn't looking at `/etc/hosts` at all. If you're using libc5, look at `/etc/resolv.conf`; it should say something like:

```
order hosts,bind
```

so your `/etc/hosts` file is checked first. If you're running GNU libc6, check your `/etc/nsswitch.conf` file. Make sure it says something like

```
hosts:  files dns
```

again, in order to make sure `/etc/hosts` is seen first.

If you have a hostname set for your machine, and this hostname does not appear in /etc/hosts, you will be able to telnet to port 25 and even send a mail with rcpt to: user@host-not-in-/etc/hosts, but fetchmail can't seem to get in touch with sendmail, no matter what you set smtpaddress to.

We had another report from a Linux user of fetchmail 2.1 who solved his SMTP connection problem by removing the reference to -lresolv from his link line and relinking. Apparently in some older Linux distributions the libc bind library version works better.

As of 2.2, the configure script has been hacked so the bind library is linked only if it is actually needed. So under Linux it won't be, and this particular cause should go away.

R2. When I try to configure an MDA, fetchmail doesn't work.

(I hear this one from people who have run into the blank-line problem in X1.)

Try sending yourself test mail and retrieving it using the command-line options '`-k -m cat`'. This will dump exactly what fetchmail retrieves to standard output (plus the Received line fetchmail itself adds to the headers).

If the dump doesn't match what shows up in your mailbox when you configure an MDA, your MDA is mangling the message. If it doesn't match what you sent, then fetchmail or something on the server is broken.

R3. Fetchmail dumps core when given an invalid rc file.

Note that this bug should no longer occur when using prepackaged fetchmail versions or installing unmodified original tarballs, since these ship with a proper parser .c file.

This is usually reported from AIX or Ultrix, but has even been known to happen on Linuxes without a recent version of `flex` installed. The problem appears to be a result of building with an archaic version of lex.

Workaround: fix the syntax of your .fetchmailrc file.

Fix: build and install the latest version of flex.

~~R4. Fetchmail dumps core in -V mode, but operates normally otherwise.~~

The information that used to be here referred to bugs in Linux libc5 systems, which are deemed obsolete by now.

R5. Running fetchmail in daemon mode doesn't work.

We have one report from a SunOS 4.1.4 user that trying to run fetchmail in detached daemon mode doesn't work, but that using the same options with -N (nodetach) is OK. We have another report of similar behavior from one Linux user, but many other Linux users report no problem.

If this happens, you have a specific portability problem with the code in daemon.c that detaches and backgrounds the daemon fetchmail. The isolated Linux case has been chased down to a failure in dup(2) that may reflect a glibc bug.

As a workaround, you can start fetchmail with -N and an ampersand to background it. A Sun user recommends this:

```
(fetchmail --nodetach <other params> &)
```

The extra pair of parens is significant --- it makes sure that the process detaches from the initial shell (one more shell is started and dies immediately, detaching fetchmail and making it child of PID 1). This is important when you start fetchmail interactively and than quit interactive shell. The line above makes sure fetchmail lives after that!

R6. Fetchmail randomly dies with socket errors.

Check the MTU value in your PPP interface reported by `/sbin/ifconfig`. If it's over 600, change it in your PPP options file. (`/etc/ppp/options` on my box). Here are option values that work:

```
mtu 552
mru 552
```

Another circumstance that can trigger this is if you are polling a virtual-mail-server name that is round-robin connected to different actual servers, so you get different IP addresses on different poll cycles. To work around this, change the poll name either to the real name of one of the servers in the ring or to a corresponding IP address.

R7. Fetchmail running as root stopped working after an OS upgrade

In RH 6.0, the HOME value in the boot-time root environment changed from /root to / as the result of a change in init. Move your .fetchmailrc or use a -f option to explicitly point at the file. (Oddly, a similar problem has been reported from Debian systems.)

R8. Fetchmail is timing out after fetching certain messages but before deleting them

There's a TCP/IP stalling problem under Redhat Linux 6.0 that can cause this symptom. Brian Boutel writes:

> TCP timestamps are turned on on my Linux boxes (I assume it's now the default). This uses 12 extra bytes per segment. When the tcp connection starts, the other end agrees a MSS of 1460, and then fragments 1460 byte chunks into 1448 and 12, because is is not allowing for the timestamp.
>
> Then, for reasons I can't explain, it waits a long time (typically 2 minutes) after the ack is sent before sending the next (fragmented) packet. Turning off tcp timestamps avoids the fragmentation and restores normal behaviour. To do this, [execute]
>
> echo 0 > /proc/sys/net/ipv4/tcp_timestamps
>
> I'm still unclear about the details of why this is happening. At least [now] I am now getting good performance and no queue blocking.

R9. Fetchmail is timing out during message fetches

This is probably a general networking issue. Sending a "RETR" command will cause the server to start sending large amounts of data, which means large packets. If your networking layer has a packet-fragmentation problem or improper firewall settings break Path MTU discovery (when for instance all ICMP traffic is blocked), that's where you'll see it.

~~R10. Fetchmail is dying with SIGPIPE.~~

Fetchmail 6.3.5 and newer block SIGPIPE, and many older versions have already handled this signal, so you shouldn't be seeing SIGPIPE at all.

R11. My server is hanging or emitting errors on CAPA.

Your POP3 server is broken. You can work around this with the declaration `auth password` in your .fetchmailrc.

R12. Fetchmail isn't working and reports getaddrinfo errors.

1. Make sure you haven't mistyped the host name or address, and that your DNS is working. If you cannot fix DNS, give the numeric host literal, for instance, 192.168.0.1
2. Make sure your `/etc/services` file (or other services database) contains the necessary service entries. If you cannot fix the services database, use the --service option and give the numeric port address. Common port addresses are:

   service port

service	port
IMAP	143
IMAP+SSL	993
POP3	110
POP3+SSL	995

R13. What does "Interrupted system call" mean?

Non-fatal signals (such as timers set by fetchmail itself) can interrupt long-running functions and will then be reported as "Interrupted system call". These can sometimes be timeouts.

R14. Since upgrading fetchmail/OpenSSL, I can no longer connect!

If the upgrade you did encompassed an upgrade to OpenSSL 1.0.0 or newer, you may need to run `c_rehash` on your certificate directories, particularly if you are using local certs directories (f. i. through fetchmail's `--sslcertpath` option).

Reason: OpenSSL 1.0.0, relative to earlier versions, uses a different hash for the symbolic links (symlinks) in its `certs/` directory, so you need to recreate the symlinks by running **`c_rehash /etc/ssl/certs`** (adjust this to where your installation keeps its certificates), and you cannot easily share this certs directory with applications linked against older OpenSSL versions.

Note: OpenSSL's `c_rehash` script is broken in several versions, which can cause malfunction if several OpenSSL tools versions are installed in parallel in separate directories. In such cases, you may need a workaround to get things going. Assuming your OpenSSL 3.0 is installed in `/opt/openssl3.0` and your certificates are in `/home/hans/certs`, you'd do this (the corresponding fetchmail option is **`--sslcertpath /home/hans/certs`** on the commandline and **`sslcertpath /home/hans/cert`** in the rcfile):

```
env PATH=/opt/openssl3.0/bin /opt/openssl3.0/bin/c_rehash /home/hans/certs
```

R15. Help, I'm getting Authorization failure!

First, try upgrading to fetchmail 6.3.18 or newer. Release 6.3.18 has received a considerable number of bug fixes for the authentication feature (AUTH, AUTHENTICATE, SASL). Most notably, fetchmail aborts SASL authentication attempts properly with an asterisk if it detects that it cannot make progress with a particular authentication scheme. This fixes issues where GSSAPI-enabled fetchmail cannot authenticate against Microsoft Exchange 2007 and 2010. **Note** that this is a bug in old fetchmail versions!

Fetchmail by default attempts to authenticate using various schemes. Fetchmail tries these schemes in order of descending security, meaning the most secure schemes are tried first.

However, sometimes the server offers a secure authentication scheme that is not properly configured, or an authentication scheme such as GSSAPI that requires credentials to be acquired externally. In some situations, fetchmail cannot know that the scheme will fail beforehand, without trying it. In most cases, fetchmail should proceed to the next authentication scheme automatically, but this sometimes does not work.

Solution: Configure the right authentication scheme explicitly, for instance, with **`--auth cram-md5`** or **`--auth password`** on the command line or `auth "cram-md5"` or `auth "password"` in the rcfile. Details can be found in the manual page.
Note that auth password should only be used across secure links (see the sslcertck and ssl/sslproto options).

Hangs and lockups

H1. Fetchmail hangs when used with pppd.

Your problem may be with pppd's 'demand' option. We have a report that fetchmail doesn't play well with it, but works with pppd if 'demand' is turned off. We have no idea why this is.

H2. Fetchmail hangs during the MAIL FROM exchange.

The symptom: 'fetchmail -v' retrieves mail fine, but appears to hang after sending the MAIL FROM command

```
SMTP> MAIL FROM: <someone@somewhere>
```

The hang is actually occuring when sendmail looks up a sender's address in DNS. The problem isn't in fetchmail but in the configuration of sendmail. You must enable the 'nodns' and 'nocanonify' features of sendmail.

Here was my fix for RedHat 7.2:

1. # cd /etc/mail
2. # cp sendmail.mc sendmail-mine.mc
3. Edit sendmail-mine.mc and add lines:

   ```
   FEATURE(nodns)
   FEATURE(nocanonify)
   ```
4. Build a new sendmail.cf

   ```
   # m4 sendmail-mine.cf > /etc/sendmail.cf
   ```
5. Restart sendmail.

For more details consult the file /usr/share/sendmail-cf/README.

H3. Fetchmail hangs while fetching mail.

Symptom: 'fetchmail -v' retrieves the first few messages, but hangs returning:

```
fetchmail: SMTP< 550 5.0.0 Access denied
fetchmail: SMTP> RSET
fetchmail: SMTP< 250 2.0.0 Reset state
.......fetchmail:  flushed
fetchmail: POP3> DELE 1
fetchmail: POP3< +OK marked deleted
```

Check and see if you're allowing sendmail connections through TCP wrappers.

Adding 'sendmail : 127.0.0.1' to /etc/hosts.allow could solve this problem.

Disappearing mail

D1. I think I've set up fetchmail correctly, but I'm not getting any mail.

Maybe you have a .forward or alias set up that you've forgotten about. You should probably remove it.

Or maybe you're trying to run fetchmail in multidrop mode as root without a .fetchmailrc file. This doesn't do what you think it should; see question C1.

Or you may not be connecting to the SMTP listener. Run fetchmail -v and see R1.

Or you may have your local user set incorrectly. In the following line

```
user 'remoteuser' there with password '*' is 'localuser' here
```

make sure that 'localuser' does exist and can receive mail.

D2. All my mail seems to disappear after a dropped connection.

One POP3 daemon used in the Berkeley Unix world that reports itself as POP3 version 1.004 actually throws the queue away. 1.005 fixed that. If you're running this one, upgrade immediately. (It also truncates long lines at column 1024.)

Many POP servers, if an interruption occurs, will restore the whole mail queue after about 10 minutes. Better ones will restore it right away. If you have an interruption and don't see it right away, cross your fingers and wait ten minutes before retrying.

Good servers are designed to restore the entire queue, including messages you have deleted. If you have one of these and it flakes out on you a lot, try setting a small `--fetchlimit` value. This will result in more IP connects to the server, but will mean it actually executes changes to the queue more often.

D3. Mail that was being fetched when I interrupted my fetchmail seems to have been vanished.

Fetchmail only sends a delete mail request to the server when either (a) it gets a positive delivery acknowledgment from the SMTP listener, or (b) it gets one of the spam-filter errors (see the description of the `antispam>` option) from the listener. No interrupt can cause it to lose mail.

However, IMAP2bis has a design problem in that its normal fetch command marks a message 'seen' as soon as the fetch command to get it is sent down. If for some reason the message isn't actually delivered (you take a line hit during the download, or your port 25 listener can't find enough free disk space, or you interrupt the delivery in mid-message) that 'seen' message can lurk invisibly in your server mailbox forever.

Workaround: add the '`fetchall`' keyword to your fetch options.

Solution: switch to an IMAP4 server.

Multidrop-mode problems

M0. What about multidrop, anyways?

Multidrop is a way to receive mail for several receivers in one mailbox. It is sort of abusing the system (using address extensions would be more useful, where available) and has quite some requirements to work properly. I wrote a separate article to document this, it is avavailable in English and in German.

M1. I've declared local names, but all my multidrop mail is going to root anyway.

Somehow your fetchmail is never recognizing the hostname part of recipient names it parses out of Envelope-header lines (or these are improperly configured) as matching a name within the designated domains. To check this, run fetchmail in foreground with -v -v on. You will probably see a lot of messages with the format "line rejected, %s is not an alias of the mailserver" or "no address matches; forwarding to %s."

These errors usually indicate some kind of configuration problem.

The easiest workaround is to add a '`via`' option (if necessary) and add enough '`aka`' declarations to cover all of your mailserver's aliases, then say '`no dns`'. This will take DNS out of the picture (though it means mail may be uncollected if it's sent to an alias of the mailserver that you don't have listed).

Occasionally these errors indicate the sort of header-parsing problem described in M7.

M2. I can't seem to get fetchmail to route to a local domain properly.

A lot of people want to use fetchmail as a poor man's internetwork mail gateway, picking up mail accumulated for a whole domain in a single server mailbox and then routing based on what's in the To/Cc/Bcc lines.

In general, this is not really a good idea. It would be smarter to just let the mail sit in the mailserver's queue and use fetchmail's ETRN or ODMR modes to trigger SMTP sends periodically (of course, this means you have to poll more frequently than the mailserver's expiration period). If you can't arrange this, try setting up a UUCP feed.

If neither of these alternatives is available, multidrop mode may do (though you *are* going to get hurt by some mailing list software; see the caveats under THE USE AND ABUSE OF MULTIDROP MAILBOXES on the man page, and check what is needed at Matthias Andree's "Requisites for working multidrop mailboxes"). If you want to try it, the way to do it is with the '`localdomains`' option.

In general, if you use localdomains you need to make sure of two other things:

1. You've actually set up your .fetchmailrc entry to invoke multidrop mode.

Many people set a '`localdomains`' list and then forget that fetchmail wants to see more than one name (or the wildcard '*') in a '`here`' list before it will do multidrop routing.

2. You may have to set 'no envelope'.

Normally, multidrop mode tries to deduce an envelope address from a message before parsing the To/Cc/Bcc lines (this enables it to avoid losing to mailing list software that doesn't put a recipient address in the To lines).

Some ways of accumulating a whole domain's messages in a single server mailbox mean it all ends up with a single envelope address that is useless for rerouting purposes. In this particular case, sell your ISP a clue. If that does not work, you may have to set '`no envelope`' to prevent fetchmail from being bamboozled by this, but a missing

envelope makes multidrop routing unreliable.

Check also answer T1 on a reliable way to do multidrop delivery if your ISP (or your mail redirection provider) is using qmail.

M3. I tried to run a mailing list using multidrop, and I have a mail loop!

This isn't fetchmail's fault. Check your mailing list. If the list expansion includes yourself or anybody else at your mailserver (that is, not on the client side) you've created a mail loop. Just chop the host part off any local addresses in the list.

If you use sendmail, you can check the list expansion with `sendmail -bv`.

~~M4. My multidrop fetchmail seems to be having DNS problems.~~

The answer that used to be here no longer applies to fetchmail.

M5. I'm seeing long DNS delays before each message is processed.

Use the '`aka`' option to pre-declare as many of your mailserver's DNS names as you can. When an address's host part matches an aka name, no DNS lookup needs to be done to check it.

If you're sure you've pre-declared all of your mailserver's DNS names, you can use the '`no dns`' option to prevent other hostname parts from being looked up at all.

Sometimes delays are unavoidable. Some SMTP listeners try to call DNS on the From-address hostname as a way of checking that the address is valid.

M6. How do I get multidrop mode to work with majordomo?

In order for sendmail to execute the command strings in the majordomo alias file, it is necessary for sendmail to think that the mail it receives via SMTP really is destined for a local user name. A normal virtual-domain setup results in delivery to the default mailbox, rather than expansion through majordomo.

Michael <michael@bizsystems.com> gave us a recipe for dealing with this case that pairs a run control file like this:

```
poll your.pop3.server proto pop3:
    no envelope no dns
    localdomains virtual.localdomain1.com virtual.localdomain2.com ...
    user yourISPusername is root * here,
    password yourISPpassword fetchall
```

with a hack on your local sendmail.cf like this:

```
##############################################
#  virtual info, local hack for ruleset 98  #
##############################################

# domains to treat as direct mapped local domain

CVvirtual.localdomain1.com virtual.localdomain2.com ...
--------------------------
in ruleset 98 add
------------------------
# handle virtual users
```

```
R$+ <@ $=V . >          $: $1 < @ $j . >
R< @ > $+ < @ $=V . >   $: $1 < @ $j . >
R< @ > $+               $: $1
R< error : $- $+ > $*   $#error $@ $1 $: $2
R< $+ > $+ < @ $+ >     $: $>97 $1
```

This ruleset just strips virtual domain names off the addresses of incoming mail. Your sendmail must be 8.8 or newer for this to work. Michael says:

> I use this scheme with 2 virtual domains and the default ISP user+domain and service about 30 mail accounts + majordomo on my inside pop3 server with fetchmail and sendmail 8.83

M7. Multidrop mode isn't parsing envelope addresses from my Received headers as it should.

It may happen that you're getting what appear to be well-formed sendmail Received headers, but fetchmail can't seem to extract an envelope address from them. There can be a couple of reasons for this.

Spurious Received lines need to be skipped:

First, fetchmail might be looking at the wrong Received header. Normally it looks only on the first one it sees, on the theory that that one was last added and is going to be the one containing your mailserver's theory of who the message was addressed to.

Some (unusual) mailserver configurations will generate extra Received lines which you need to skip. To arrange this, use the optional skip prefix argument of the 'envelope' option; you may need to say something like '`envelope 1 Received`' or '`envelope 2 Received`'.

The 'by' clause doesn't contain a mailserver alias:

When fetchmail parses a Received line that looks like

```
Received: from send103.yahoomail.com (send103.yahoomail.com [205.180.60.92])
    by iserv.example.net (8.8.5/8.8.5) with SMTP id RAA10088
    for <ksturgeon@fbceg.example.org>; Wed, 9 Sep 1998 17:01:59 -0700
```

it checks to see if 'iserv.example.net' is a DNS alias of your mailserver before accepting 'ksturgeon@fbceg.example.org' as an envelope address. This check might fail if your DNS were misconfigured, or if you were using 'no dns' and had failed to declare iserv.example.net as an alias of your server. The typical hint is logging similar to: `line rejected, iserv.example.net is not an alias of the mailserver`, if you use fetchmail in verbose mode.

Workaround: You can specify the alias explicitly, with `aka` *`iserv.example.net`* statements in the rcfile. Replace *iserv.example.net* by the name you find in **your** 'by' part of the 'Received:' line.

M8. Users are getting multiple copies of messages.

It's a consequence of multidrop. What's happening is that you have N users subscribed to the same list. The list software sends N copies, not knowing they will end up in the same multidrop box. Since they are both locally addressed to all N users, fetchmail delivers N copies to each user.

Fetchmail tries to eliminate adjacent duplicate messages in a multidrop mailbox. However, this logic depends on the message-ID being identical in both copies. It also depends on the two copies being adjacent in the server mailbox. The former is usually the case, but the latter condition sometimes fails in a timing-dependent way if the server was processing multiple incoming mail streams.

I could eliminate this problem by keeping a list of all message-IDs received during a poll so far and dropping any message that matches a seen mail ID. The trouble is that this is an O(N**2) operation that might significantly slow down the retrieval of large mail batches.

The real solution however is to make sure that fetchmail can find the envelope recipient properly, which will reliably prevent this message duplication.

Mangled mail

X1. Spurious blank lines are appearing in the headers of fetched mail.

What's probably happening is that the POP/IMAP daemon on your mailserver is inserting a non-RFC822 header (like X-POP3-Rcpt:) and something in your delivery path (most likely an old version of the *deliver* program, which sendmail often calls to do local delivery) is failing to recognize it as a header.

This is not fetchmail's problem. The first thing to try is installing a current version of *deliver*. If this doesn't work, try to figure out which other program in your mail path is inserting the blank line and replace that. If you can't do either of these things, pick a different MDA (such as maildrop) and declare it with the '`mda`' option.

X2. My mail client can't see a Subject line.

First, see X1. This is quite probably the same problem (X-POP3-Rcpt header or something similar being inserted by the server and choked on by an old version of *deliver*).

The O'Reilly sendmail book does warn that IDA sendmail doesn't process X- headers correctly. If this is your problem, all I can suggest is replacing IDA sendmail, because it's broken and not RFC822 conformant.

X3. Messages containing "From" at the start of line are being split.

If you know the messages aren't split in your server mailbox, then this is a problem with your POP/IMAP server, your client-side SMTP listener or your local delivery agent. Fetchmail cannot split messages.

Some POP server daemons ignore Content-Length headers and split messages on From lines. We have one report that the 2.1 version of the BSD popper program (as distributed on Solaris 2.5 and elsewhere) is broken this way.

You can test this. Declare an mda of 'cat' and send yourself one piece of mail containing "From" at start of a line. If you see a split message, your POP/IMAP server is at fault. Upgrade to a more recent version.

Sendmail and other SMTP listeners don't split RFC822 messages either. What's probably happening is either sendmail's local delivery agent or your mail reader are not quite RFC822-conformant and are breaking messages on what it thinks are Unix-style From headers. You can figure out which by looking at your client-side mailbox with vi or more. If the message is already split in your mailbox, your local delivery agent is the problem. If it's not, your mailreader is the problem.

If you can't replace the offending program, take a look at your sendmail.cf file. There will likely be a line something like

```
Mlocal, P=/usr/bin/procmail, F=lsDFMShP, S=10, R=20/40, A=procmail -Y -d $u
```

describing your local delivery agent. Try inserting the 'E' option in the flags part (the F= string). This will make sendmail turn each dangerous start-of-line From into a >From, preventing programs further downstream from acting up.

X4. My mail is being mangled in a new and different way

The first thing you need to do is pin down what program is doing the mangling. We don't like getting bug reports about fetchmail that are actually due to some other program's malfeasance, so please go through this diagnostic sequence before sending us a complaint.

There are five possible culprits to consider, listed here in the order they pass your mail:

1. Programs upstream of your server mailbox.
2. The POP or IMAP server on your mailserver host.
3. The fetchmail program itself.
4. Your local sendmail.
5. Your LDA (local delivery agent), as called by sendmail or specified by `mda`.

Often it happens that fetchmail itself is OK, but using it exposes pre-existing bugs in your downstream software, or your downstream software has a bad interaction with POP/IMAP. You need to pin down exactly where the message is being garbled in order to deduce what is actually going on.

The first thing to do is send yourself a test message, and retrieve it with a .fetchmailrc entry containing the following (or by running with the equivalent command-line options):

```
mda "cat >MBOX" keep fetchall
```

This will capture what fetchmail gets from the server, except for (a) the extra Received header line fetchmail prepends, (b) header address changes due to `rewrite`, and (c) any end-of-line changes due to the `forcecr` and `stripcr` options. MBOX will in fact contain what programs downstream of fetchmail see.

The most common causes of mangling are bugs and misconfigurations in those downstream programs. If MBOX looks unmangled, you will know that is what is going on and that it is not fetchmail's problem. Take a look at the other FAQ items in this section for possible clues about how to fix your problem.

If MBOX looks mangled, the next thing to do is compare it with your actual server mailbox (if possible). That's why you specified `keep`, so the server copy would not be deleted. If your server mailbox looks mangled, programs upstream of your server mailbox are at fault. Unfortunately there is probably little you can do about this aside from complaining to your site postmaster, and nothing at all fetchmail can do about it!

More likely you'll find that the server copy looks OK. In that case either the POP/IMAP server or fetchmail is doing the mangling. To determine which, you'll need to telnet to the server port and simulate a fetchmail session yourself. This is not actually hard (both POP3 and IMAP are simple, text-only, line-oriented protocols) but requires some attention to detail. You should be able to use a fetchmail -v log as a model for a session, but remember that the "*" in your LOGIN or PASS command dump has to be replaced with your actual password.

The objective of manually simulating fetchmail is so you can see exactly what fetchmail sees. If you see a mangled message, then your server is at fault, and you probably need to complain to your mailserver administrators. However, we like to know what the broken servers are so we can warn people away from them. So please send us a transcript of the session including the mangling *and the server's initial greeting line*. Please tell us anything else you think might be useful about the server, like the server host's operating system.

If your manual fetchmail simulation shows an unmangled message, congratulations. You've found an actual fetchmail bug, which is a pretty rare thing these days. Complain to us and we'll fix it. Please include the session transcript of your manual fetchmail simulation along with the other things described in the FAQ entry on reporting bugs.

~~X5. Using POP3, retrievals seems to be fetching too much!~~

The information that used to be here pertained to fetchmail 4.4.7 or older, which should not be used. Use a recent fetchmail version.

X6. My mail attachments are being dropped or mangled.

Fetchmail doesn't discard attachments; fetchmail doesn't have any idea that attachments are there. Fetchmail treats the body of each message as an uninterpreted byte stream and passes it through without alteration. If you are not receiving attachments through fetchmail, it is because your mailserver is not sending them to you.

The fix for this is to replace your mailserver with one that works. If its operating system makes this difficult, you should replace its operating system with one that works. Windows- and NT-based POP servers seem especially prone to mangle attachments. If you are running one of these, replacing your server with a Unix machine is probably the only effective solution.

We've had sporadic reports of problems with Microsoft Exchange and Outlook servers. These sometimes randomly fail to ship attachments to your client. This is a known bug, acknowledged by Microsoft.

They may also mangle the attachments they do pass through. If you see unreadable attachments with a ContentType of "application/x-tnef", you're having this problem. The TNEF utility may help.

The Mail Max POP3 server and the InterChange and Imail IMAP servers are known to simply drop MIME attachments when uploading messages.

We've also had a report that Lotus Notes sometimes trashes the MIME type of messages. In particular, it seems to modify MIME headers of type application/pdf, mangling the type to application/octet-stream. It may corrupt other MIME types as well.

The IMAP service of Lotus Domino has a known bug in the way it generates MIME Content-type headers (observed on Lotus Domino 5.0.2b). It's a subtle one that doesn't show up when Netscape Messenger and other clients use a FETCH BODY[] to grab the whole message. When fetchmail uses FETCH RFC822.HEADER and FETCH RFC822.TEXT to get first the header and then the body, Domino generates different Boundary tags for each part, e.g. one tag is declared in the Content-type header and another is used to separate the MIME parts in the body. This doesn't work. (I have heard a rumor that this bug is scheduled to be fixed in Domino release 6; you can find a workaround at contrib/domino.)

Rob Funk explains: Unfortunately there also remain many mail user agents that don't write correct MIME messages. One big offender is Sun MailTool attachments, which are formatted enough like MIME that some programs could get confused; these are generated by the mailtool and dtmail programs (the mail programs in Sun's OpenWindows and CDE environments).

One solution to problems related to misformatted MIME attachments is the emil program; see its tutorial file at that site for details on emil. It is useful for converting character sets, attachment encodings, and attachment formats. At this writing, emil does not appear to have been maintained since a patch to version 2.1.0beta9 in late 1997, but it is still useful.

One good way of using emil is from within procmail. You can have procmail look for signs of problematic message formatting, and pipe those messages through emil to be fixed. emil will not always be able to fix the problem, in which case the message is unchanged.

A possible rule to be inserted into a .procmailrc file for using emil would be:

```
:0HB
* 1^1 ^Content-Type: \/X-sun[^;]*
* 1^1 ^Content-Type: \/application/mac-binhex[^;]*
* 1^1 ^Content-Transfer-Encoding: \/x-binhex[^;]*
* 1^1 ^Content-Transfer-Encoding: \/x-uuencode[^;]*
{
  LOG="Converting $MATCH
"
  :0fw
  | emil -A B -T Q -B BA -C iso-8859-1 -H Q -F MIME \
  | gawk '{gsub(/\r\n?/,"\n");print $0}'
}
```

The "1^1" in the conditions is a way of specifying to procmail that if any one of the four listed expressions is found in the message, the total condition is considered true, and the message gets passed into emil. These four subconditions check whether the message has a Sun attachment, a binhex attachment, or a uuencoded attachment; there are others that could be added to check these things better and to check other relevant conditions. The

"LOG=" line writes a line into the procmail log; the lone double-quote beginning the following line makes sure the log entry gets an end-of-line character. The call to gawk (GNU awk) is for fixing end-of-line conventions, since emil sometimes leaves those in the format of the originating machine; it could probably be replaced with a sed subsitution.

The emil call itself tries to ensure that the message uses:

- BinHex encoding for any Apple Macintosh-only attachments
- Quoted-Printable encoding for text (when necessary)
- Base64 Encoding for binary attachments
- iso-8859-1 character set for text (unfortunately emil can't yet convert from windows-1252 to iso-8859-1)
- Quoted-Printable encoding for headers
- MIME attachment format

Most of these (the primary exceptions being the character set and the Apple binary format) are as they should be for good internet interoperability.

Some mail servers (Lotus Domino is a suspect here) mangle Sun-formatted messages, so the conversion to MIME needs to happen before such programs see the message. The ideal is to rid the world of Sun-formatted messages: don't use mailtool for sending attachments (it doesn't understand MIME anyway, and most of the world doesn't understand its attachments, so it really shouldn't be used at all), and make sure dtmail is set to use MIME rather than mailtool's format.

X7. Some mail attachments are hanging fetchmail.

Fetchmail doesn't know anything about mail attachments and doesn't treat them any differently from plain message data.

The most usual cause of this problem seems to be bugs in your network transport layer's capability to handle the very large TCP/IP packets that attachments tend to turn into. You can test this theory by trying to download the offending message through a webmail account; using HTTP for the message tends to simulate large-packet stress rather well, and you will probably find that the messages that seem to be choking fetchmail will make your HTTP download speed drop to zero.

This problem can be caused by subtle bugs in the packet-reassembly layer of your TCP/IP stack; these often don't manifest at normal packet sizes. It may also be caused by malfunctioning path-MTU discovery on the mailserver. Or, if there's a modem in the link, it may be because the attachment contains the Hayes mode escape "+++".

X8. A spurious) is being appended to my messages.

Fetchmail 6.3.5 and newer releases are supposed to fix this.

Due to the problem described in S2, the IMAP support in fetchmail cannot follow the IMAP protocol 100 %. Most of the time it doesn't matter, but if you combine it with an SMTP server that behaves unusually, you'll get a spurious) at the message end.

One piece of software that can trigger this is the Interchange mail server, as used by, e.g., mailandnews.com. Here's what happens:

1. Someone sends mail to your account. The last line of the message contains text. So at the SMTP level, the message ends with, e.g. "blahblah\r\n.\r\n"
2. The SMTP handler sees the final "\r\n.\r\n" and recognizes the end of the message. However, instead of doing the normal thing, which is tossing out the ".\r\n" and leaving the first '\r\n' as part of the email body, Interchange throws out the whole "\r\n.\r\n", and leaves the email body without any line terminator at the end of it. RFC821 does not forbid this, though it probably should.
3. Fetchmail, or some other IMAP client, asks for the message. IMAP returns it, but it's enclosed inside

parentheses, according to the protocol. The message size in bytes is also present. Because the message doesn't end with a line terminator, the IMAP client sees:

```
....blahblah)...
```

where the ')' is from IMAP.

4. Fetchmail only deals with complete lines, and can't trust the stated message size because Microsoft Exchange goofs it up.
5. As a result, fetchmail takes the final 'blahblah)' and puts it at the end of the message it forwards on. If you have verbosity on, you'll get a message about actual != expected.

X9. Missing "Content-Transfer-Encoding" header with Domino IMAP

Domino 6 IMAP was found by Anthony Kim in February 2006 to erroneously omit the "Content-Transfer-Encoding" header in messages downloaded through IMAP, causing messages to display improperly. This happened with Domino's incoming mail format configured to "Prefers MIME". Solution: switch Domino to "Keep in Sender's format".

Reference: Anthony Kim's list post

X10. Fetchmail delivers partial messages

Fetchmail is sometimes reported to deliver partial messages. This is usually related to network outages that occur while fetchmail is downloading a message body. In such cases, fetchmail has downloaded a complete header, so your header will be intact. The message body will be truncated, and fetchmail will later attempt to redownload the message (providing the server is standards conformant).

The reason for the truncation is that fetchmail streams the body directly from the POP3/IMAP server into the SMTP/LMTP server or MDA (in order to save memory), so fetchmail has already written a part of the message before it notices it will be incomplete, and fetchmail cannot abort a transaction it has started, and it's unclear if it ever will be able to, because this is not standardized and the outcome will depend on the receiving software (be it SMTP/LMTP or MDA).

Other problems

O1. The --logfile option doesn't work if the logfile doesn't exist.

This is a feature, not a bug. It's in line with normal practice for system daemons and allows you to suppress logging by removing the log file, without hacking potentially fragile startup scripts. To get around it, just touch(1) the logfile before you run fetchmail (this will have no effect on the contents of the logfile if it already exists).

O2. Every time I get a POP or IMAP message, the header is dumped to all my terminal sessions.

Fetchmail uses the local sendmail to perform final delivery, which Mozilla and other clients don't do; the announcement of new messages is done by a daemon that sendmail pokes. There should be a "biff" command to control this. Type

```
biff n
```

to turn it off. If this doesn't work, try the command

```
chmod -x $(tty)
```

which is essentially what `biff -n` will do. If this doesn't work, comment out any reference to "comsat" in your /etc/inetd.conf file and reload (or restart) inetd.

In Slackware Linux distributions, the last line in /etc/profile is

```
biff y
```

Change this to

```
biff n
```

to solve the problem system-wide.

O3. Does fetchmail reread its rc file every poll cycle?

No, but versions 5.2.2 and later will notice when you modify your rc file and restart, reading it. Note that this causes troubles if you need to provide a password via the console, unless you're running in --nodetach mode.

O4. Why do deleted messages show up again when I take a line hit while downloading?

According to the POP3 RFCs, deletes aren't actually performed until you issue the end-of-session QUIT command. Fetchmail cannot fix this, but there is a workaround: use the --expunge option with a reasonably low figure that works for you. Try 10 for a start.

IMAP is less susceptible to this problem, because the "deleted" message marks are persistent, but they aren't in POP3. Note that the --expunge default for IMAP is different than the default for POP3.

If you get very unlucky, you might take a line hit in the window between the delete and the expunge. If you've set a longer expunge interval, the window gets wider. This problem should correct itself the next time you complete a successful query.

O5. Why is fetched mail being logged with my name, not the real From address?

Because logging is done based on the address indicated by the sending SMTP's MAIL FROM, and some listeners are picky about that address.

Some SMTP listeners get upset if you try to hand them a MAIL FROM address naming a different host than the originating site for your connection. This is a feature, not a bug -- it's supposed to help prevent people from forging mail with a bogus origin site. (RFC 1123 says you shouldn't do this exclusion...)

Since the originating site of a fetchmail delivery connection is localhost, this effectively means these picky listeners will barf on any MAIL FROM address fetchmail hands them with an @ in it!

Versions 2.1 and up try the header From address first and fall back to the calling-user ID. So if your SMTP listener isn't picky, the log will look right.

O6. I'm seeing long sendmail delays or hangs near the start of each poll cycle.

Sendmail does a hostname lookup when it first starts up, and also each time it gets a HELO in listener mode.

Your resolver configuration may be causing one of these lookups to fail and time out. Check your `/etc/resolv.conf`, `/etc/host.conf`, `/etc/nsswitch.conf` (if you have the latter two) and you `/etc/hosts` files. Make sure your hostname and fully-qualified domain name are both in `/etc/hosts`, and that hosts is looked at before DNS is queried. You probably also want your remote mail server(s) to be in the hosts file.

You can suppress the startup-time lookup if need to by reconfiguring with `FEATURE(nodns)`.

Configuring your bind library to cache DNS lookups locally may help, and is a good idea for speeding up other services as well. Switching to a faster MTA like Postfix might help.

O7. Why doesn't fetchmail deliver mail in date-sorted order?

Because that's not the order the server hands it to fetchmail in.

Fetchmail getting mail from a POP server delivers mail in the order that your server delivers mail. Fetchmail can't do anything about this; it's a limitation of the underlying POP protocol.

In theory it might be possible for fetchmail in IMAP mode to sort messages by date, but this would be in violation of two basics of fetchmail's design philosophy: (a) to be as simple and transparent a pipe as possible, and (b) to *hide*, rather than emphasize, the differences between the remote-fetch protocols it uses.

Re-ordering messages is a user-agent function, anyway.

O8. I'm using pppd. Why isn't my monitor option working?

There is a combination of circumstances that can confuse fetchmail. If you have set up demand dialing with pppd, and pppd has an idle timeout, and you have lcp-echo-interval set, then the lcp-echo-interval time must be longer than the pppd idle timeout. Otherwise it is going keep increasing the packet counters that fetchmail relies upon, triggering fetchmail into polling after its own delay interval and thus preventing the pppd link from ever reaching its inactivity timeout.

O9. Why does fetchmail keep retrieving the same messages over and over?

First, check to see that you haven't enabled the *keep* and *fetchall* option. If you have, turn one of them off - which one, depends on why they have been set in the first place, and to a lesser degree on the upstream server.

This can also happen when some other mail client is logged in to your mail server, if it uses a simple exclusive-locking scheme (and many, especially most POP3 servers, do exactly that). Your fetchmail is able to retrieve the messages, but because the mailbox is write-locked by the other instance yours can neither mark messages seen or delete them. The solution is to either (a) wait for the other client to finish, or (b) terminate it.

~~O10. Why is the received date on all my messages the same?~~

The answer that used to be here made no sense and was dropped.

O11. I keep getting messages that say "Repoll immediately" in my logs.

This is your server barfing on the CAPA probe that fetchmail sends. Because some servers like to drop the connection after that probe, fetchmail will re-poll immediately with this probe defeated.

If you run fetchmail in daemon mode (say "set daemon 600"), you will get the message only once per run.

If you set an authentication method explicitly (say, with `auth password`), you will never get the message.

O12. Fetchmail no longer expunges mail on a 451 SMTP response.

This is a feature, not a bug.

Any 4xx response (like 451) indicates a transient (temporary) error. This means that the mail could be accepted if retried later. Lookup failures are normally transient errors as a mail should not get rejected if a dns server is unreachable or down.

A permanent reject response is of the form 5xx (like 550).

You could tell your SMTP server to not lookup any addresses if you are not keen on checking the sender addresses. This problem typically occurs if your mail server is not checking the sender addresses, but your local server is.

Or you could declare `antispam 451`, which is not recommended though, as it may cause mail loss.

Or, you could check your nameserver configuration and query logs for dns errors.

All these issues are not related to fetchmail directly.

O13. I want timestamp information in my fetchmail logs.

Write a `preconnect` command in your configuration file that does something like "date >> $HOME/fetchmail.log".

O14. Fetchmail no longer deletes oversized mails with --flush.

Use `--limitflush` (available since release 6.3.0) to delete oversized mails along with the `--limit` option. If you are already having `flush` in your rcfile to delete oversized mails, *replace* it with `limitflush` to avoid losing mails unintentionally.

The `--flush` option is primarily designed to delete mails which have been read/downloaded but not deleted yet. This option cannot be overloaded to delete oversized mails as it cannot be guessed whether the user wants to delete only read/downloaded mails or only oversized mails or both when a user specifies both `--limit` and `--flush`. Hence, a separate `--limitflush` has been added to resolve the ambiguity.

O15. Fetchmail always retains the first message in the mailbox.

This happens when fetchmail sees an "X-IMAP:" header in the very first message in your mailbox. This usually stems from a message like the one shown below, which is automatically created on your server. This message shows up if the University of Washington IMAP or PINE software is used on the server together with a POP2 or POP3 daemon that is not aware of these messages, such as some versions of Qualcomm Popper (QPOP):

```
From MAILER-DAEMON Wed Nov 23 11:38:42 2005
Date: 23 Nov 2005 11:38:42 +0100
From: Mail System Internal Data <MAILER-DAEMON@imap.example.org>
Subject: DON'T DELETE THIS MESSAGE -- FOLDER INTERNAL DATA
Message-ID: <1132742322@imap.example.org>
X-IMAP: 1132742306 0000000001
Status: RO

This text is part of the internal format of your mail folder, and is not
a real message.  It is created automatically by the mail system software.
If deleted, important folder data will be lost, and it will be re-created
with the data reset to initial values.
```

As this message does not contain useful information, fetchmail is not retrieving it. And deleting it might slow down the server if you are keeping messages on the server, and the server would recreate it anyways, that's why fetchmail does not bother to delete it either.

O16. Why is the Fetchmail FAQ only available in ISO-216 A4 format? How do I get the FAQ in Letter format?

All the world uses ISO-216:1975 "A4" paper except for North America. Using A4 format reaches far more people than (formerly known as DIN A4, from DIN 476) format. Besides that, A4 paper *is* available in North America. For further information on the Letter-vs-A4 story, see:

- Markus Kuhn: "International standard paper sizes"
- Brian Forte: "A4 vs US Letter"

Offering the document formatted for two different paper sizes would bloat the package beyond reason, and formatting in a way that fits A4 and Letter paper formats would be a waste of paper in most parts of the world. For that reason, fetchmail only ships with an A4 formatted PDF document.

To create a letter-sized PDF, install HTMLDOC, edit `fetchmail-FAQ.book` in the source directory with your favorite text editor, replace `--size A4` by `--size letter`, and type:

```
make fetchmail-FAQ.pdf
```

O17. Linux logs "TCP(fetchmail:...): Application bug, race in MSG_PEEK."

That's in fact a bug in Linux kernels around the late 2.6.2X versions, rather than fetchmail. Fetchmail has no race bugs around MSG_PEEK, as of version 6.3.9. The message can safely be ignored.

Back to Fetchmail Home Page $Date$

Eric S. Raymond <esr@thyrsus.com>

Matthias Andree