Sendmail

Release 2.0, 22nd March 2004.

Contents

Chapter 1

Some sendmail History

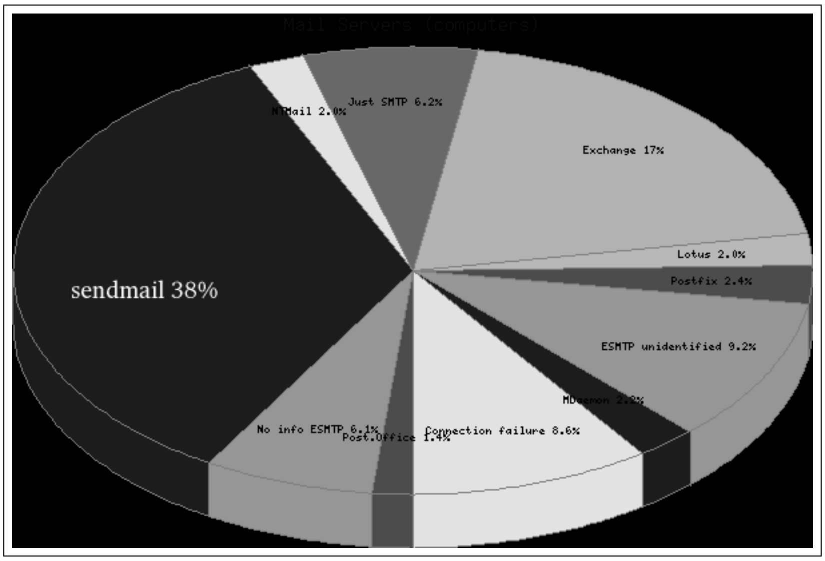

Figure 1.1: The percentage of `.com.au` MX records which responded with matchable greetings for various common mail servers

1.1 What is sendmail?

- The main mail transfer agent on the internet
- First release called "sendmail" in 1983.
- Is the default mail service on most versions of Unix

Notes...

Dan Bernstein (of `qmail` fame) found that somewhere between 50% and 75% of all email servers ran sendmail.

My study in 2001 of Australian MX records showed sendmail to handle far more domains than any other mail server. I queried every `.com.au` domain that I could find mentioned on `www.netcraft.com` and saved the response from the greeting. I then pattern-matched these against known responses from various mail servers. Figure 1.1 gives the proportions of responses I received, with *sendmail* far, far in the lead. And of course, this doesn't count the many *sendmail* administrators who choose to change the greeting to something different.

Most large ISPs run sendmail, including AOL, Compuserve. The major alternatives to sendmail are Postfix (`www.postfix.org`), qmail (`www.qmail.org`), exim (`www.exim.org`) and ZMailer (`www.zmailer.org`).

A much smaller number of organisations receive their internet email directly to either Microsoft's Exchange System, IBM's Lotus Notes system, or Novell's Groupwise system.

1.2 Sendmail's good features

- handles high loads well
- is extremely configurable to handle legacy protocols

Notes. . .

sendmail is quite simple and efficient, and so can handle many times more simultaneous messages than more complex systems such as Microsoft's Exchange System or IBM's Lotus Notes system. Also, it is quite easy to tell *sendmail* to stop listening for new messages if the CPU load is getting too high, preventing small problems running away into big ones.

The delivery mechanisms that *sendmail* uses (which we will get to in chapter 6) are very flexible. It can be configured to send messages using any kind of arbitrary program. This lets *sendmail* route messages for any kind of protocol or medium that tools exist for.

1.3 Sendmail's bad features

- had an administrator-unfriendly configuration file
- a history of security problems
- a lot of legacy influences

Notes. . .

For much of this course we will be looking at the `sendmail.cf` file. The format of this file has been likened to line noise and swear words from comic strips. Fortunately though, in real life you don't need to edit it all that often – normally you would generate a configuration from the very simple `.mc` file. We will cover this in chapter 9.

Before version 8.9, SPAM relaying was allowed by default. For the previous 15 years, this was seen as reasonable – the internet was about academic co-operation, so forwarding email on behalf of someone else was seen as appropriate and good-netizen behaviour.

Sendmail versions 8.6 and before had security problems of a fairly significant nature. For example, up until that version there was an extension to the SMTP command-set called "debug" which let any remote user run any command (as root) on the mail server.

In recent times security problems have not been quite so dramatic. For example, in October 2002 it was discovered that `smrsh` (the restricted *sendmail* shell) wasn't quite as restricted as was previously thought. A clever user could subvert `smrsh` to run any program on the system that the user had privileges to run. It would not allow a user to escalate privileges like many of the other security problems of the past.

1.4 Sendmail versions

v8.9 anti-spam

v8.10 Mail filter API

v8.11 LDAP, SMTP authentication, transport security

v8.12 no longer SUID root

Notes. . .

1.5 What version am I running?

- `telnet localhost 25`
- `echo ’$Z’ |/usr/sbin/sendmail -bt -d0`

Notes. . .

By default, sendmail will greet remote connections with its version number (and patch level).

Alternatively, you can ask sendmail to do a rule test with debugging information and just give it the input “$Z”. (Which means “the configuration file version number.)

To query a binary on your local host, the following command should display its version number, along with some extra configuration information, possibly including the configuration version number:

```
echo ’$Z’ |sendmail -bt -d0

Version 8.12.1
Compiled with:  MAP_REGEX LOG MATCHGECOS MIME7TO8 MIME8TO7 NAMED_BIND
NETINET NETUNIX NEWDB NIS QUEUE SCANF SMTP USERDB XDEBUG

============ SYSTEM IDENTITY (after readcf) ============
(short domain name) $w = forest
(canonical domain name) $j = forest.ifost.org.au
(subdomain name) $m = ifost.org.au
(node name) $k = forest.ifost.org.au
========================================================

ADDRESS TEST MODE (ruleset 3 NOT automatically invoked)
Enter <ruleset> <address>
> 8.12.1
```

1.6 Exercise

What version are you running?

Notes...

1. Use the two techniques from page 13.

Chapter 2

On the wire protocols

2.1 Method 1: Message Injection Protocol

- Runs on port 587
- Can do authentication of the end user
- Message may get rewritten
- Documented in rfc2476
- Otherwise, same as Method 2
- New (v8.11) – rarely used.

Notes. . .

From `http://www.sendmail.net/rfc2476.net`:

> Despite its intimidating title, RFC 2476 doesn't propose a whole new type of service with its own new protocol. Rather, RFC 2476 describes how the usual protocols for SMTP service should be tightened up at the point where mail enters the system, rather than being routed from one site to another. It proposes a new standard for a message submission agent (MSA), designed to replace the more general-purpose mail transfer agent (MTA) as the first service to which a mail user agent (MUA) connects to deliver a mail message.
>
> The goal? First, to prevent spammers and unauthorized users from launching messages into your Internet mail system by tightening up that first conversation between, say, Eudora and sendmail. An MSA would require more fully formed headers to make authentication and tracking of the message possible. It may have extra error codes, such as "message violates system policy." It may require authentication (see RFC 2554) before talking to the MUA.
>
> Second, to separate the mail submission function from mail transfer and relay. As Claus Assmann says, "Whereas an MSA will 'repair' and add headers as necessary, an MTA shouldn't do anything with headers unless it's acting as a gateway to a different email protocol. So if we can cleanly separate both functions, the MTA can stop doing 'header munging' - for instance, host name canonification."
>
> The RFC's authors say their goals are fivefold:
>
> 1. **Implement security policies and guard against unauthorized mail relaying or injection of unsolicited bulk mail.** Translation: Stop spammers before they start.
> 2. **Implement authenticated submission, including off-site submission by authorized users such as travelers.** That is, use a password or digital ID to let your sales team send mail from outside the firewall, while blocking strangers.

3. **Separate the relevant software code differences, thereby making each code base more straightforward and allowing for different programs for relay and submission.** We want to separate mail submission from mail transfer as a function, just as mail routing and mail hosting have been separated in the past. And we don't want to make everyone update their working mail transfer agents (MTAs) if they could just add a MSA at the front end instead. With an established, stable code base, sendmail 8.10 can serve as either MSA or MTA, depending on its configuration.
4. **Detect configuration problems with a site's mail clients.** Whereas you wouldn't want to log errors and page an admin for errors at other sites, you would want to know when your own users were having trouble submitting mail from their desks. This is another reason to create an MSA that's distinct from the MTA.
5. **Provide a basis for adding enhanced submission services in the future.** Because security and spam are both cops-and-robbers games requiring constant improvements.

2.2 Method 2: SMTP on port 25

- Greet with **HELO**
- Announce the sender with **MAIL FROM:**
- Say who should receive it with **RCPT TO:**
- Send email body after **DATA**
- Finish with **.**
- Terminate with **QUIT**

Notes. . .

Here is a sample exchange; the connecting party is in boldface.

```
telnet mail.company.com 25
220 mail.company.com ESMTP Sendmail 8.8.8 ready at Thu, 1 Apr 2038
HELO mail.sender.net
250 mail.company.com Hello mail.sender.net [1.2.3.4], pleased to meet you.
MAIL FROM: me@sender.net
250 me@sender.net Sender...  ok
RCPT TO: person@company.com
250 person@company.com...  Recipient ok
DATA
354 Enter mail, end with "." on a line by itself.

From: me@sender.net
To: person@company.com
Cc: boss@sender.net
Subject: Sample message
Date: Thu, 1 Apr 2038 11:59:20 +1000 (EST)
I hope you enjoyed receiving
this sample message.
.
250 OAAA19247 Message accepted for delivery
QUIT
221 mail.company.com closing connection.
```

2.3 Odd things about SMTP

1. The `From:` field in the **DATA** doesn't have to match the **MAIL FROM**
2. Neither does `To:` have to match **RCPT TO**
3. None of the header fields need to exist.

Notes. . .

This of course makes it very easy to fake email messages to appear to come from someone else. The Win32.BugBear virus was one of the first viruses to do this to make it harder to identify the infected computer.

2.4 Exercise

Talking SMTP manually. . .

Notes. . .

1. Connect to your neighbour's machine and fake a message manually to a user on that system. (If there are no ordinary user accounts on your neighbour's system, get them to create one.)
2. You should be able to read it by logging in as that user and running `mail`.

2.5 How spooled e-mail gets delivered

- The mail server does a DNS lookup for MX records for the domain name
- It gets several names back, each with a priority number
- Try delivering to the smallest number
- If it fails, it tries the next lowest
- If everything fails, try again later.
- Warn after 4 hours. Give up after 4 days.

Notes. . .

The domain name is easy to work out, it's just the portion after @ in the email address.

Suppose we are trying to send an email to `person@mvac.nsw.edu.au`. The mail server looks up the MX record for `mvac.nsw.edu.au` "10 mail.mvac.nsw.edu.au" and "65535 postoffice.telstra.net". So it tries to connect to port 25 on `mail.mvac.nsw.edu.au` (a cheap-bandwidth satellite link) and then talk SMTP to it, or `postoffice.telstra.net` if something goes wrong.

2.6 Exercise

Walking through the lookup process...

Notes...

1. If you run `echo just a test |mail someuser@neighbourscomputer` does an email message arrive?
2. Try running `host -t mx company.com`, replacing "company.com" with one of your friend's email address. If you are on an older version of Unix or on Windows NT/2k, you might need to run `nslookup -type=mx company.com` instead.

Chapter 3

The configuration file

3.1 sendmail.cf

- Only read at startup/SIGHUP time
- **Solaris** `/etc/mail/sendmail.cf`
  HP-UX `/etc/mail/sendmail.cf`
  ***BSD** `/etc/mail/sendmail.cf`
  Linux `/etc/sendmail.cf`
- Some versions of Unix still “freeze” it to a `sendmail.fc`

Notes. . .

If you start *sendmail* with a full path name (e.g. /usr/sbin/sendmail), then you can tell it to reconfigure itself by sending it the HUP signal.

```
kill -HUP ‘cat /var/run/sendmail.pid‘
```

Unixes that require a freeze step usually do this with `sendmail -fc`.

Sendmail is not too fussy about its state. It copes quite happily with being `kill`ed and started manually. Remember that `ps -ef | grep sendmail` will find the PID of the process on most Unixes; `ps -waux | grep sendmail` does the same job on BSD unixes.

Normally sendmail gets run as `sendmail -bd -q30m`

-bd means “become daemon”;

-q30m means “retry messages that could not be sent every 30 minutes”.

Of course, on SYSV Unixes, the easiest thing to remember is the run script commands (`/etc/rc.d/init.d/sendmail stop; /etc/rc.d/init.d/sendmail start`). HP-UX and older versions of SuSE Linux use `/sbin/init.d/sendmail start`.

3.2 Configuration file format

- 14 different options
- Blank lines
- Comments begin with “#”
- Lines beginning with tab carry on from the previous line.

Notes. . .

D something for later use, usually a macro

C a class (a set) for use in a rule, with the elements of the set defined on this line

F a class from the contents of a file

K a map/program to look up values with

R define a rule in the current ruleset

S introduce a new ruleset number

O set some kind of option

H define a header that needs rewriting

T define who the trusted users are

V what version of configuration file layout this is

P precedence definitions

M define a method of delivery

Q queue definitions

X milter (mail filter) definitions

3.3 Some easy things to change

DS A smart relay host

DM What domain to masquerade as

Dj My hostname

O SmtpGreetingMessage What banner to give on connection.

Notes. . .

The complete list of macros and options available in *sendmail*is in chapter 14 on page 176.

Here are some common definitions:

- `DS`smart.relay.host
- `DM`domain.net
- `Dj`mail.ifost.org.au
- `O SmtpGreetingMessage=$j` MAIL-SERVER `$b`

Note that the way an option is set (with an equals sign) is not the same as the way a macro definition is made.

Also notice that D macros get used in three different ways:

- We define it using `Dj`
- The variables is called "`j`"
- We use it later (in other definitions, for example) by using `$j`

Keep in mind that if you change the rules (see chapter 4), you could be using any of these `D` variables for anything you feel like. What is described here is just the normal conventions for these variables.

`DS` and `DM` are normally blank. `DS` will set a "smart relay" through which all email is passed on to for handling - this has to be a hostname, not an IP address. `DM` ("masquerade domain") requests that sendmail makes any outgoing message appear to have come from the domain listed[1].

HP-UX machines are often set with only a short hostname. Whenever sendmail starts up it will try first to get a fully-qualified hostname, which will fail by default. On these systems it is necessary to set `Dj`.

[1] Actually, it won't do this for *every* outgoing message; there are several other variables that determine which messages get masqueraded.

`O SmtpGreetingMessage` used to be configured with `De`. (On HP-UX systems, it still is.) This defines the greeting text sent when another system connects to port 25. Many security experts recommend changing the default sendmail greeting banner to something more obscure which does not include the version number of the software.

3.4 Exercise

Modifying the sendmail configuration file...

Notes...

1. Backup your current `sendmail.cf` to somewhere safe. You may need to copy it back in place many times in this course if you make too many mistakes!

2. Find the `O SmtpGreetingMessage` option (or `De` macro if you are using an older version of sendmail). Change it to something else. Restart sendmail, try `telnet yourserver 25` and confirm that that it has worked.

   According to RFC821, the mail server should greeting with its own name as the first word. (And then say ESMTP or SMTP, which sendmail will put in place for you.) This means that `SmtpGreetingMessage` often begins with `$j`.

3. Set `DM` to the domainname for your favourite overseas company. Log in to your machine as an ordinary user (not *root*) and send an email message from your machine to some other machine. What email address does it appear to come from?

   Note that there may be no `DM` line at all. If so, you will need to insert a line somewhere (e.g. with the other macro definitions) to define it.

4. Work with your neighbour (or with an empty machine). Set up a user *user1* on both systems (if it is not already created). One of you should set `DS` to the name of your neighbour's system and then restart sendmail. From that same machine, send an email to *user1* – on which machine does it arrive?

Challenge question The SmtpGreetingMessage includes two variables `$v` (sendmail version) and `$Z` (configuration file version). Why are these two variables separate?

Chapter 4

Rewriting Rules

4.1 What is a ruleset?

- A “subroutine” for rewriting an address
- Can get applied to a source address
- Can get applied to a destination address
- Can get called from other rulesets
- Order doesn’t matter
- Is defined by `Sname` and then lots of `R...` lines

Notes...

Historically, all rules in sendmail used to be just numbers. So all ruleset started with `S0`, `S1` ...`S99`. Names were introduced in v8.9.

Now it is possible to have the one ruleset be given a name and a number. For example, ruleset number 0 often has the name “parse”. You will see this in `sendmail.cf`as `Sparse=0`

4.2 What import rule sets are there?

canonify=3 All addresses

Parse=0 How to send?

1 Process sender address

2 Process recipient address

final=4 Postprocess all addresses

localaddr=5 Rewrite unaliased

check_relay, check_mail, check_rcpt, check_compat Is this sender allowed to go to this recipient through our machine?

Notes...

We will cover this again at the end of chapter 6.

4.3 Rewrites rule OK

1. **R**left hand side tokens *tab* replacements
2. **R**left hand side tokens *tab* **$:** replacements
3. **R**left hand side tokens *tab* **$@** final result
4. **R**left hand side tokens *tab* **$#** delivery mechanism, host and user

Notes. . .

The general idea is that the email address will be checked against the left hand side; if it matches, it will be replaced by the right hand side.

The first rewrite rule is a loop that will keep on trying. So if it matches successfully, it will be tried again and again until it stops matching.

The second rewrite rule only operates once and then moves on to the next rewrite rule in the ruleset. This is the most common kind of operation, so you will see this quite often.

The third rewrite rule operates once and *returns from the ruleset*, skipping all the remaining rules. This is for the case “I have found turned the email address into the format I need and do not need to do anything more”.

The fourth rewrite rule operates only once and tells `sendmail` what *delivery mechanism* to use. We will discuss delivery mechanisms in chapter 6

4.4 Things on the left hand side

`$|` Meta-separator

$* Match zero or more tokens

$+ Match one or more tokens

$- Exactly one token

$=x Match any phrase in class x

$˜x Match any word not in class x

$@ Match nothing

Notes...

What is a meta-separator?

Remember that a ruleset may be called by another ruleset, like a subroutine called from a main function. Sometimes they want to return several values (e.g. the tidied up email address, plus, “yes, relaying is OK”.) For plenty of examples of this, have a look at the `Rcpt_ok` ruleset.

To do that kind of trick, we need to have some one of separating the two values. But we can’t use any ordinary character, because that might be part of a genuine email address[1]. So we need a character that does not exist in the ASCII character set to use as a separator[2]. This is `$|`.

What are tokens and operators?

The other $ forms are more comprehensible. $*, $+ and $- are similar to their regular expression counterparts, except they work on tokens.

Tokens are defined as “things that are separated by operators”. The set of operator characters is defined by the `OperatorChars` option. By default, this is set up something like this:

```
O OperatorChars=.:@!^/[]+%
```

But regardless of what `OperatorChars` says the characters `()<>,;` are always operators as well.

By default underscore (_) is not in `OperatorChars`, but period / fullstop (`.`) is. This means that a rewrite rule that works perfectly well for handling names such as `greg_baker` will not necessarily copy-and-paste to work for `greg.baker`.

[1] Remember, email addresses are not just `user@domain.name`. All sorts of other addresses are possible, such as `computer1!computer2!user` or worse.

[2] As it turns out, `$|`is character 155. There was a bug in Solaris 8 that meant that charcter 155 was sometimes equal to character 105 (the letter “i”). This produced bizarre results to say the least!

What is a class?

We'll get to that in chapter 5.

4.5 Things on the right hand side

$*n* The *n*th thing that was matched on the left

$[*name*$] Canonicalize *name*

$(*map key* $@*arguments* $:*default* $) Find *key* in *map*, otherwise *default*

$>*n* Call ruleset *n* with the rest of the line

Letters, symbols, numbers, $| Just substitute it

Notes. . .

What is a map lookup?

We'll get to that in chapter 5.

4.6 More things on the right hand side

Extra TAB Everything following is a comment

$#*mechanism* $@ *host* $: *user* Only in ruleset `0` or `check_rcpt` or similar

$#error $@ *number* $: *error string* Die with the error given (including SMTP error code number)

Notes. . .

4.7 Example Ruleset

```
Sappend_domainname
R$@ tab $#error $@ 5.7.1 $:  "550 Arrgh"
R$* @ $* tab $@ $1 @ $2
R$* tab $:  $1 @ ifost.org.au
```

Notes. . .

The first rule handles the case if `append_domainname` was given nothing at all as an argument.

The second rule says “if there is already a domain name on the end of this address, just return it unchanged.

The third rule adds “@ifost.org.au” on to the end of the any address that we are left with. The $: is very important – otherwise it will keep appending “@ifost.org.au” an infinite number of times!

4.8 How to test

```
sendmail -bt
ADDRESS TEST MODE (ruleset 3 NOT automatically invoked)
Enter <ruleset> <address>
> append_domainname gregb
append_domainnam input:  gregb
append_domainnam returns:  gregb @ ifost .  org .  au
```

Notes. . .

It's probably best not to modify `sendmail.cf`in place. Copy it somewhere else (e.g. `mysample.cf`, make your changes and then run:

```
sendmail -bt -Cmysample.cf
```

There is a handy debugging option `-d21.12` which will make *sendmail* print out the workspace after every single rule is processed.

4.9 Exercises

Making your own rewrites...

Notes...

1. Put the rules from page 4.7 into your `sendmail.cf`. Now test it with a few examples.
2. Take the $: out from the example. Try again with something domainless. What happens?
3. Do users at your site have email addresses of this form[3]?

   `firstname_lastname@yourcompany.com`

   What happens when someone sends an email to `firstname.lastname@yourcompany.com`? Write a rewrite rule that could be used to turn the "." into "_".
4. Does your company have both a `.com` and a `.com.au` domain name? Make a rewrite rule that turns all your `.com.au` addresses into `.com` ones.

[3] Well, it's an exercise. If they don't, pretend they do.

Chapter 5

Classes and Maps

5.1 Class definitions

- `CM`
- `Cw localhost loghost`
- `FR-o /etc/mail/relay-domains`

Notes. . .

Classes are like sets. They have elements in them, and we can ask whether something is found in the set.

The first line aboves defines an empty class `M`. When we write a rewrite rule, nothing will match `M`. `M` is traditionally used to indicate which domains should be masqueraded. (They get masqueraded to the value of `$M` normally.)

The second line defines a class `w` with two elements in it: `localhost` and `loghost`. `w` is traditionally used to help work out whether an email address is on this computer or not. It should be a list of the different names this computer is known by (including different aliases you have for `localhost`).

The last line defines a class `R`. Instead of being listed within `sendmail.cf`, *sendmail* is instructed to look in `/etc/mail/relay-domains`. It expects to find a file with several lines in it – each line will be an element in the class `R`.

`R` is very significant for mail hubs. It lists the names of computers that we are willing to relay for. If this file is empty then no client machines will be able to use it as a SMTP mail server – it will simply reject anything that comes to it.

The `-o` option tells *sendmail* that it is optional. If the file is not found, to carry on as if the file were there, but empty. (i.e. the class will be the empty set.)

5.2 Some examples

```
From Canonify2=96:
 R$* < @ $=M > $*  tab  $:  $1 < @ $2 .  > $3
From Relay_ok
R$=R $*  tab  $@ RELAY  tab  relayable IP address
```

Notes. . .

Note the syntax for matching an element in a class – `$=R`.

The `Canonify2` ruleset is trying to make sure that every domainname ends in a "."[1] Anything that we are going to masquerade doesn't need any significant processing, so we just put a dot on the end and leave it at that.

Why the < and >, you may ask? Because the email addresses we are handling here are of the form:

`Greg Baker < gregb@ifost.org.au >`

The `Relay_ok` ruleset returns the word "RELAY" if this is appropriate for relaying. It is given just the canonicalised hostname of the computer trying to send the message.

[1] Or alternatively, it could append the local domainname. e.g. I have a message with a sender of `user@nsw`. Presumably that should be turned into `user@nsw.mycompany.com` or something like that, while `user@recently-acquired.com` should be left alone.

5.3 Other class tricks

- `FL/etc/passwd %[^:]`
- `Fg |/some/program`

Notes. . .

Sendmail can be compiled to allow a `scanf(3)` string on the `F` line. This lets you do simplistic parsing of text files. The first example above lets you read all the user names on your system out of `/etc/passwd` file into a class `L`.

Of course, if you want to do something much cleverer, you can make a class out of the output of a program that gets run when sendmail starts up. It can't be given any command line arguments – if you want some, make a little script file for it. This is what the second example is doing. The possibilities for this are only limited by your imagination:

- A class for all the hosts listed in `/etc/hosts`
- A class from the GECOS field in `/etc/passwd`
- A class from a database of virtual hosts.
- . . .

5.4 Exercise

Playing with classes...

Notes...

1. Your company has a small subsidiary. The eight staff there use the same mail server as your "ordinary" staff, but their email addresses are `user@subsidiary.com`. Create two rewrite rules – one that will translate those eight users from `user@yourcompany.com` to `user@subsidiary.com` and another to go back again.

2. The following script will take the GECOS field out of `/etc/passwd`, lowercase it and replace spaces with underscores. i.e. if the GECOS field has "John Smith" it will become "john_smith".

   ```
   cut -d:  -f5 /etc/passwd | tr ’[:upper:]  ’  ’[:lower:]_’
   ```

   Now, suppose your users want to have email addresses like this:

   `firstname_lastname@yourcompany.com`.

   Write a rewrite rule that will respond "YES" if the given email address matches a user based on the GECOS field.

3. (Optional) Extend the previous example. You also want to have email addresses like this:

   `firstname.lastname@yourcompany.com`

   A simple way of solving this problem is to take the script from the previous question and change the final underscore (_) to a dot (.).

   Could you use a rewrite rule to mangle these into the same format as the first form?

5.5 Problems with classes

- Only read at *sendmail* startup time
- Can only copy unchanged to the the right hand side
- A little inflexible

Notes. . .

This is not to say that classes are not useful for a lot of static information. But for most complicated data, what we really want are things called *maps*. . .

5.6 What is a map?

A lookup from something to something else:

- a username → GECOS field
- Query DNS, NIS/NIS+ or LDAP
- Find an entry in a flat file or indexed file
- A regular expression
- Run a program with an argument

Notes...

dbm Database lookups using the ndbm(3) library. *sendmail*must be compiled with **NDBM** defined.

btree Database lookups using the btree interface to the Berkeley DB library. *sendmail* must be compiled with **NEWDB** defined.

hash Database lookups using the hash interface to the Berkeley DB library. *sendmail* must be compiled with **NEWDB** defined.

nis NIS lookups. *sendmail* must be compiled with **NIS** defined.

nisplus NIS+ lookups. *sendmail* must be compiled with **NISPLUS** defined. The argument is the name of the table to use for lookups, and the **-k** and **-v** flags may be used to set the key and value columns respectively.

hesiod Hesiod lookups. *sendmail* must be compiled with **HESIOD** defined.

ldap LDAP X500 directory lookups. *sendmail*must be compiled with **LDAPMAP** defined. The map supports most of the standard arguments and most of the command line arguments of the *ldapsearch* program. Note that, by default, if a single query matches multiple values, only the first value will be returned unless the **-z** (value separator) map flag is set. Also, the **-1** map flag will treat a multiple value return as if there were no matches.

netinfo NeXT NetInfo lookups. *sendmail*must be compiled with **NETINFO** defined.

text Text file lookups. The format of the text file is defined by the **-k** (key field number), **-v** (value field number), and **-z** (field delimiter) flags.

-z can be a letter or the special strings ”\n” or ”\t” (for newline and tab respectively). If left blank, it will assume a sequence of whitespace.

ph PH query map. Contributed and supported by Mark Roth, `roth@uiuc.edu`. For more information, consult the web site `www-dev.cso.uiuc.edu/sendmail`

nsd nsd map for IRIX 6.5 and later. Contributed and supported by Bob Mende of SGI, `mende@sgi.com`.

stab Internal symbol table lookups. Used internally for aliasing.

implicit Really should be called "alias" – this is used to get the default lookups for alias files, and is the default if no class is specified for alias files.

user Looks up users using *getpwnam (3).* The **-v** flag can be used to specify the name of the field to return (although this is normally used only to check the existence of a user).

host Canonifies host domain names. Given a host name it calls the name server to find the canonical name for that host.

bestmx Returns the best MX record for a host name given as the key. The current machine is always preferred - that is, if the current machine is one of the hosts listed as a lowest-preference MX record, then it will be guaranteed to be returned. This can be used to find out if this machine is the target for an MX record, and mail can be accepted on that basis. If the **-z** flag is given, then all MX names are returned, separated by the given delimiter.

dns This map requires the option -R to specify the DNS resource record type to lookup. The following types are supported: A, AAAA, AFSDB, CNAME, MX, NS, PTR, SRV, and TXT. A map lookup will return only one record. Hence for some types, e.g., MX records, the return value might be a random element of the list due to randomizing in the DNS resolver.

sequence The arguments on the 'K' line are a list of maps; the resulting map searches the argument maps in order until it finds a match for the indicated key. For example, if the key definition is:

```
Kmap1 ...
Kmap2 ...
Kseqmap sequence map1 map2
```

then a lookup against "seqmap" first does a lookup in map1. If that is found, it returns immediately. Otherwise, the same key is used for map2.

syslog the key is logged via *syslogd(8).* The lookup returns the empty string.

switch Much like the "sequence" map except that the order of maps is determined by the service switch. The argument is the name of the service to be looked up; the values from the service switch are appended to the map name to create new map names. For example, consider the key definition:

```
Kali switch aliases
```

together with the service switch entry:

```
aliases nis files
```

This causes a query against the map "ali" to search maps named "ali.nis" and "ali.files" in that order.

dequote Strip double quotes (") from a name. It does not strip backslashes, and will not strip quotes if the resulting string would contain unscannable syntax (that is, basic errors like unbalanced angle brackets; more sophisticated errors such as unknown hosts are not checked). The intent is for use when trying to accept mail from systems such as DECnet that routinely quote odd syntax such as

"49ers::ubell"

A typical usage is probably something like:

```
Kdequote dequote
...
R$- tab $: $(dequote $1 $) R$- $+ tab $: $>3 $1 $2
```

Care must be taken to prevent unexpected results; for example,

"|someprogram < input > output"

will have quotes stripped, but the result is probably not what you had in mind. Fortunately these cases are rare.

regex The map definition on the **K** line contains a regular expression. Any key input is compared to that expression using the POSIX regular expressions routines regcomp(), regerr(), and regexec(). Refer to the documentation for those routines for more information about the regular expression matching. No rewriting of the key is done if the **-m** flag is used. Without it, the key is discarded or if **-s** if used, it is substituted by the substring matches, delimited by **$|**or the string specified with the the **-d** flag. The flags available for the map are

-n not

-f case sensitive

-b basic regular expressions (default is extended)

-s substring match

-d set the delimiter used for -s

-a append string to key

-m match only, do not replace/discard value

-D perform no lookup in deferred delivery mode.

The **-s** flag can include an optional parameter which can be used to select the substrings in the result of the lookup. For example,

-s1,3,4

Notes: to match a **$** in a string, \$$ must be used. If the pattern contains spaces, they must be replaced with the blank substitution character, unless it is space itself.

program The arguments on the **K** line are the pathname to a program and any initial parameters to be passed. When the map is called, the key is added to the initial parameters and the program is invoked as the default user/group id. The first line of standard output is returned as the value of the lookup. This has many potential security problems, and has terrible performance; it should be used only when absolutely necessary.

macro Set or clear a macro value. To set a macro, pass the value as the first argument in the map lookup. To clear a macro, do not pass an argument in the map lookup. The map always returns the empty string. Example of typical usage include:

```
Kstorage macro
......
# set macro ${MyMacro} to the ruleset match
R$+ tab $: $(storage {MyMacro} $@ $1 $) $1
# set macro ${MyMacro} to an empty string
R$* tab $: $(storage {MyMacro} $@ $) $1
# clear macro ${MyMacro}
R$- $: $(storage {MyMacro} $) $1
```

arith Perform simple arithmetic operations. The operation is given as key, currently +, -, *, /, %, |, & (bitwise OR, AND), l (for less than), and = are supported. The two operands are given as arguments. The lookup returns the result of the computation, TRUE or FALSE for comparisons, integer values otherwise. All options which are possible for maps are ignored. A simple example is:

```
Kcomp arith
...
Scheck_etrn
R$* tab $: $(comp l $@ $&load_avg $@ 7 $) $1
RFALSE $# error ...
```

5.7 How do I use a hash map?

1. Create a text file `myfile`
2. Make a hash from it `/etc/stuff`
3. Put `Kmyhash /etc/stuff` in `sendmail.cf`
4. Make a rule `R$*` *tab* `$(myhash $1 $)`

Notes. . .

Hash maps are by far the most common, and probably the simplest of the efficient ones.

- `vi myfile`
- `makemap hash /etc/mail/mymap < myfile`

  This actually creates a file `/etc/mail/mymap.db`
- `editmap -q hash /etc/mail/mymap somekey`

  Just check that the `makemap` worked sensibly! The `-q` option means "query".
- `vi /etc/mail/sendmail.cf`

  Put a line like this in `sendmail.cf` to create the map:

  `Kmystuff hash /etc/mail/mymap`

  Note, there is no `.db` on the end of this. Now, make a rule that uses it.

  `R$*` *tab* `$(mystuff $1 $)`
- Restart *sendmail*, or send it a HUP signal.
- If you want to alter the contents of the hash, you can use
  - `editmap hash /etc/mail/mymap somekey newvalue`

    This just changes the one key "somekey" to have the value "newvalue".
  - `makemap -o ...`

    Rather than recreating the map from scratch, this will append to an existing map.
  - `makemap -o -r ...`

    Same idea, but it doesn't complain about replacing existing entries.
  - `makemap ...`

    Start again from scratch!

5.8 Exercise

Making and using simple hash maps...

Notes...

1. Your company has just acquired the domainnames *yourcompany.net*, *yourcompany.net.au*, *yourcompany.com*, *yourcompany.com.au* and *yourcompany.au.com*.

   Your boss is hinting that many more domains like that may be bought in the future. All emails in the `.au` range need to end up at your Sydney mailserver; others go to Washington. Write a rewrite rule that will take an email address with any of these endings and turn it into the right one for your organisation.

2. You are in the process of changing your DNS and mail servers. As a sanity check, you want to make sure that there is always a secondary mail server for every domain that you handle. Make two maps – a `syslog` one and also one using `bestmx -z,`. Write a ruleset that will report to `syslog` on each mail message that arrives for a domain that doesn't have a secondary MX server.

5.9 Common special flags

-a*x* append *x* for successful matches

-T*x* append *x* for temporary failures

-o Optional

-h,-b LDAP server hostname / basename

Notes. . .

Here is the complete list of them, from the *sendmail*operations guide:

-o Indicates that this map is optional - that is, if it cannot be opened, no error is produced, and *sendmail* will behave as if the map existed but was empty.

-N, -O If neither **-N** or **-O** are specified, *sendmail* uses an adaptive algorithm to decide whether or not to look for null bytes on the end of keys. It starts by trying both; if it finds any key with a null byte it never tries again without a null byte and vice versa. If **-N** is specified it never tries without a null byte and if **-O** is specified it never tries with a null byte. Setting one of these can speed matches but are never necessary. If both **-N** and **-O** are specified, *sendmail* will never try any matches at all – that is, everything will appear to fail.

-a*x* Append the string *x* on successful matches. For example, the default *host* map appends a dot on successful matches.

-T*x* Append the string *x* on temporary failures. For example, *x* would be appended if a DNS lookup returned `‘‘server failed’’` or an NIS lookup could not locate a server. See also the **-t** flag.

-f Do not fold upper to lower case before looking up the key.

-m Match only (without replacing the value). If you only care about the existence of a key and not the value (as you might when searching the NIS map `hosts.byname` for example), this flag prevents the map from substituting the value. However, The -a argument is still appended on a match, and the default is still taken if the match fails.

-k*keycol* The key column name (for NIS+) or number (for text lookups). For LDAP maps this is an LDAP filter string in which %s is replaced with the literal contents of the lookup key and %0 is replaced with the LDAP escaped contents of the lookup key according to RFC2254.

-v*valcol* The value column name (for NIS+) or number (for text lookups). For LDAP maps this is the name of one or more attributes to be returned; multiple attributes can be separated by commas. If not specified, all attributes found in the match will be returned.

-z*delim* The column delimiter (for text lookups). It can be a single character or one of the special strings "\n" or "\t" to indicate newline or tab respectively. If omitted entirely, the column separator is any sequence of white space. For LDAP maps this is the separator character to combine multiple values into a single return string. If not set, the LDAP lookup will only return the first match found.

-t Normally, when a map attempts to do a lookup and the server fails (e.g., *sendmail* couldn't contact any name server; this is *not* the same as an entry not being found in the map), the message being processed is queued for future processing. The **-t** flag turns off this behavior, letting the temporary failure (server down) act as though it were a permanent failure (entry not found). It is particularly useful for DNS lookups, where someone else's misconfigured name server can cause problems on your machine. However, care must be taken to ensure that you don't bounce mail that would be resolved correctly if you tried again. A common strategy is to forward such mail to another, possibly better connected, mail server.

-D Perform no lookup in deferred delivery mode. This flag is set by default for the *host* map.

-S*spacesub* The character to use to replace space characters after a successful map lookup (esp. useful for regex and syslog maps).

-s*spacesub* For the dequote map only, the character to use to replace space characters after a successful dequote.

-q Don't dequote the key before lookup.

-L*level* For the syslog map only, it specifies the level to use for the syslog call.

-A When rebuilding an alias file, the **-A** flag causes duplicate entries in the text version to be merged. For example, two entries:

```
list: user1, user2
list: user3
```

would be treated as though it were the single entry

```
list: user1, user2, user3
```

in the presence of the **-A** flag.

Some additional flags are available for the host and dns maps:

-d delay: specify the resolver's retransmission time interval (in seconds).

-r retry: specify the number of times to retransmit a resolver query. .pp The following additional flags are present in the ldap map only:

-R Do not auto chase referrals. sendmail must be compiled with **-DLDAP_REFERRALS** to use this flag.

-n Retrieve attribute names only.

-V*sep* Retrieve both attributes name and value(s), separated by *sep* .

-r*deref* Set the alias dereference option to one of never, always, search, or find.

-s*scope* Set search scope to one of base, one (one level), or sub (subtree).

-h*host* LDAP server hostname. Some LDAP libraries allow you to specify multiple, space-separated hosts for redundancy. In addition, each of the hosts listed can be followed by a colon and a port number to override the default LDAP port.

-b*base* LDAP search base.

-p*port* LDAP service port.

-l*timelimit* Time limit for LDAP queries.

-Z*sizelimit* Size (number of matches) limit for LDAP queries.

-d*distinguished_name* The distinguished name to use to login to the LDAP server.

-M*method* The method to authenticate to the LDAP server. Should be one of **LDAP_AUTH_NONE**, **LDAP_AUTH_SIMPLE**, or **LDAP_AUTH_KRBV4**.

-P*passwordfile* The file containing the secret key for the **LDAP_AUTH_SIMPLE** authentication method or the name of the Kerberos ticket file for **LDAP_AUTH_KRBV4**.

-1 Force LDAP searches to only succeed if a single match is found. If multiple values are found, the search is treated as if no match was found.

5.10 Classy maps

- F{VirtHosts}@ldap:-k (&(objectClass=virtHosts)(host=*)) -v host
- F{MyClass}foo@hash:/etc/mail/classes

Notes. . .

It's a little rare (I haven't seen it that I can remember), but it is possible to make a class from a map, as in the above two examples.

Chapter 6

Delivery Mechanisms

6.1 M sent me

```
Mprocmail, Path=/usr/local/bin/procmail,
Flags=mSDFMhun, S=11, R=21, Argv=procmail -m $h $g $u
```

Notes. . .

Note that you only need to give the first letter to each of the options. So you are more likely to see a `sendmail.cf` file with lines in it like this:

```
Mlocal, P=/usr/libexec/mail.local, F=lsDFMAw5:/|@qrmn9S, S=EnvFromL/HdrFromL,
R=EnvToL/HdrToL, T=DNS/RFC822/X-Unix, A=mail -d $u
```

The complete range of options are:

Path	The pathname of the mailer
Flags	Special flags for this mailer
Sender	Rewriting set(s) for sender addresses
Recipient	Rewriting set(s) for recipient addresses
recipients	Maximum number of recipients per connection
Argv	An argument vector to pass to this mailer
Eol	The end-of-line string for this mailer
Maxsize	The maximum message length to this mailer
maxmessages	The maximum message deliveries per connection
Linelimit	The maximum line length in the message body
Directory	The working directory for the mailer
Userid	The default user and group id to run as
Nice	The `nice(2)` increment for the mailer
Charset	The default character set for 8-bit characters
Type	Type information for DSN diagnostics
topWait	The maximum time to wait for the mailer
Queuegroup	The default queue group for the mailer
/	The root directory for the mailer

6.2 Flags

- Expand aliases?
- Hidden dot method?
- "/" means file or X500?
- Email addresses with comments?

Notes. . .

The following flags may be set in the mailer description. Any other flags may be used freely to conditionally assign headers to messages destined for particular mailers. Flags marked with †are not interpreted by the *sendmail* binary; these are the conventionally used to correlate to the flags portion of the **H** line. Flags marked with ‡apply to the mailers for the sender address rather than the usual recipient mailers.

a Run Extended SMTP (ESMTP) protocol (defined in RFCs 1869, 1652, and 1870). This flag defaults on if the SMTP greeting message includes the word `ESMTP` .

A Look up the user part of the address in the alias database. Normally this is only set for local mailers.

b Force a blank line on the end of a message. This is intended to work around some stupid versions of /bin/mail that require a blank line, but do not provide it themselves. It would not normally be used on network mail.

c Do not include comments in addresses. This should only be used if you have to work around a remote mailer that gets confused by comments. This strips addresses of the form "Phrase <address>" or "address (Comment)" down to just "address".

C‡ If mail is *received* from a mailer with this flag set, any addresses in the header that do not have an at sign "@" after being rewritten by ruleset three will have the "@domain" clause from the sender envelope address tacked on. This allows mail with headers of the form:

> From: usera@hosta To: userb@hostb, userc

to be rewritten as:

> From: usera@hosta To: userb@hostb, userc@hosta

automatically. However, it doesn't really work reliably.

d Do not include angle brackets around route-address syntax addresses. This is useful on mailers that are going to pass addresses to a shell that might interpret angle brackets as I/O redirection. However, it does not protect against other shell metacharacters. Therefore, passing addresses to a shell should not be considered secure.

D† This mailer wants a “Date:” header line.

e This mailer is expensive to connect to, so try to avoid connecting normally; any necessary connection will occur during a queue run. See also option **HoldExpensive**.

E Escape lines beginning with “From” in the message with a ‘>’ sign.

f This mailer wants a **-f** *from* flag, but only if this is a network forward operation (i.e., the mailer will give an error) if the executing user does not have special permissions).

F †This mailer wants a “From:” header line.

g Normally, *sendmail* sends internally generated email (e.g., error messages) using the null return address as required by RFC 1123. However, some mailers don’t accept a null return address. If necessary, you can set the **g** flag to prevent *sendmail* from obeying the standards; error messages will be sent as from the MAILER-DAEMON (actually, the value of the `$n` macro).

h Upper case should be preserved in host names (the $@ portion of the mailer triplet resolved from ruleset `0`) for this mailer.

i Do User Database rewriting on envelope sender address.

I This mailer will be speaking SMTP to another *sendmail* – as such it can use special protocol features. This flag should not be used except for debugging purposes because it uses **VERB** as SMTP command.

j Do User Database rewriting on recipients as well as senders.

k Normally when *sendmail* connects to a host via SMTP, it checks to make sure that this isn’t accidently the same host name as might happen if *sendmail* is misconfigured or if a long-haul network interface is set in loopback mode. This flag disables the loopback check. It should only be used under very unusual circumstances.

K Currently unimplemented. Reserved for chunking.

l This mailer is local (i.e., final delivery will be performed).

L Limit the line lengths as specified in RFC821. This deprecated option should be replaced by the **L=** mail declaration. For historic reasons, the **L** flag also sets the **7** flag.

m This mailer can send to multiple users on the same host in one transaction. When a `$u` macro occurs in the *argv* part of the mailer definition, that field will be repeated as necessary for all qualifying users. Removing this flag can defeat duplicate supression on a remote site as each recipient is sent in a separate transaction.

M †This mailer wants a "Message-Id:" header line.

n Do not insert a UNIX-style "From" line on the front of the message.

o Always run as the owner of the recipient mailbox. Normally *sendmail* runs as the sender for locally generated mail or as *daemon* (actually, the user specified in the **u** option) when delivering network mail. The normal behavior is required by most local mailers, which will not allow the envelope sender address to be set unless the mailer is running as daemon. This flag is ignored if the **S** flag is set.

p Use the route-addr style reverse-path in the SMTP "MAIL FROM:" command rather than just the return address; although this is required in RFC821 section 3.1, many hosts do not process reverse-paths properly. Reverse-paths are officially discouraged by RFC 1123.

P †This mailer wants a "Return-Path:" line.

q When an address that resolves to this mailer is verified (SMTP VRFY command), generate 250 responses instead of 252 responses. This will imply that the address is local.

r Same as **f**, but sends a **-r** flag.

R Open SMTP connections from a "secure" port. Secure ports aren't (secure, that is) except on UNIX machines, so it is unclear that this adds anything. *sendmail* must be running as root to be able to use this flag.

s Strip quote characters (" and \) off of the address before calling the mailer.

S Don't reset the userid before calling the mailer. This would be used in a secure environment where *sendmail* ran as root. This could be used to avoid forged addresses. If the **U=** field is also specified, this flag causes the effective user id to be set to that user.

u Upper case should be preserved in user names for this mailer. Standards require preservation of case in the local part of addresses, except for those address for which your system accepts responsibility.

U This mailer wants UUCP-style "From" lines with the ugly "remote from <host>" on the end.

w The user must have a valid account on this machine, i.e., *getpwnam* must succeed. If not, the mail is bounced. See also the **MailBoxDatabase** option. This is required to get ".forward" capability.

x †This mailer wants a "Full-Name:" header line.

X This mailer wants to use the hidden dot algorithm as specified in RFC821; basically, any line beginning with a dot will have an extra dot prepended (to be stripped at the other end). This insures that lines in the message containing a dot will not terminate the message prematurely.

z Run Local Mail Transfer Protocol (LMTP) between *sendmail* and the local mailer. This is a variant on SMTP defined in RFC 2033 that is specifically designed for delivery to a local mailbox.

Z Apply DialDelay (if set) to this mailer.

0 Don't look up MX records for hosts sent via SMTP/LMTP. Do not apply **FallbackMXhost** either.

1 Don't send null characters ('\0') to this mailer.

2 Don't use ESMTP even if offered; this is useful for broken systems that offer ESMTP but fail on EHLO (without recovering when HELO is tried next).

3 Extend the list of characters converted to =XX notation when converting to Quoted-Printable to include those that don't map cleanly between ASCII and EBCDIC. Useful if you have IBM mainframes on site.

5 If no aliases are found for this address, pass the address through ruleset 5 for possible alternate resolution. This is intended to forward the mail to an alternate delivery spot.

6 Strip headers to seven bits.

7 Strip all output to seven bits. This is the default if the **L** flag is set. Note that clearing this option is not sufficient to get full eight bit data passed through *sendmail* . If the **7** option is set, this is essentially always set, since the eighth bit was stripped on input. Note that this option will only impact messages that didn't have 8→7 bit MIME conversions performed.

8 If set, it is acceptable to send eight bit data to this mailer; the usual attempt to do 8→7 bit MIME conversions will be bypassed.

9 If set, do *limited* 7\8 bit MIME conversions. These conversions are limited to text/plain data.

: Check addresses to see if they begin ":include:" ; if they do, convert them to the "*include*" mailer.

| Check addresses to see if they begin with a '|'; if they do, convert them to the "prog" mailer.

/ Check addresses to see if they begin with a '/'; if they do, convert them to the "*file*" mailer.

@ Look up addresses in the user database.

% Do not attempt delivery on initial recipient of a message or on queue runs unless the queued message is selected using one of the -qI/-qR/-qS queue run modifiers or an ETRN request.

6.3 Special mailers

local Delivery locally

error Refuse to send

discard Silently drop the message

prog Deliver via program

file Deliver to a file

smtp/esmtp/smtp8/esmtp8 TCP protocols

relay Forward to something else

procmail Delivery via `procmail`

Notes. . .

sendmail will complain if there is no `local` delivery mechanism defined. This makes some sense really!

`error` you have already seen (for example on page 37). You do not define this, it just always exists. Likewise, for `discard`.

`prog` is quite common; the Path argument is usually `smrsh` – the *sendmail* restricted shell.

The other delivery mechanisms are not guaranteed to exist, but generally do. The difference between `smtp` and `smtp8` is that the latter can send 8-bit characters, rather than being constrained to 7-bit characters (which is more normal).

6.4 Message flow Part 1

- Recipient address tidied by ruleset `3`
- Remember this tidy result
- Which mailer to use from ruleset `0`
- . . .

Notes. . .

The tidied-up result gets used twice. First it is passed through `0`. . .

Ruleset `0` has to return with a `$#` line somewhere.

6.5 Message flow Part 2

- Send the tidy address through `2`
- Send the result through the delivery agent's `R=` ruleset
- Ruleset `4`

Notes. . .

As you can see the tidied-up rule from the previous page gets run through a few more rules before we are finished.

6.6 Message Flow Part 3

Take the *sender* address through rulesets:

- `3`
- `1`
- The delivery agent's `S=` ruleset
- `4`

Notes...

If any of these rulesets don't exist, they just get silently skipped.

6.7 Message Flow Part 4

Check flags (`F=`) on mailer:

- `A` flag? Check recipient `/etc/mail/aliases.db`
- `5` flag and alias didn't work? Try ruleset `5`
- `w` flag? Try recipient `.forward` file

Run the mailer!

Notes...

Which mailer to use? Normally the one decided by ruleset `0`, but if ruleset `5` returns `$#`, then we will start again.

Actually, if ruleset `0` makes the username `$:` portion of its response begin with an "@", then ruleset `5` never gets run. (And the "@" is silently dropped.)

6.8 About aliases

- Normally in `/etc/mail/aliases.db`
- In DBM format (usually)
- Created by running `newaliases` manually
- `newaliases` reads from `/etc/mail/aliases`
- Should redirect *postmaster* and any system accounts.

Notes. . .

The location is defined by the `AliasFile` option, which is specified `without` the `.db` ending. i.e.

```
O AliasFile=/etc/mail/aliases
```

Actually, there's much more to say about aliases. . .

6.9 Exercises

Confusing users terribly...

Notes...

1. Your users are forwarding emails to their Hotmail email accounts. Put a stop to this by modifying ruleset `0` – anything with a destination including `hotmail.com` should be delivered with the `error` mailer[1]. Check that it works by sending some emails.

2. Your company took over Victim Corporation last year. Some people are still sending emails to and from `victim.com` email addresses. Modify ruleset `3` to rewrite any such addresses. Check that it works.

3. Take the `A` flag out of your `local` mailer. Do aliases get expanded any more? What about `w`; does forwarding work?

[1] If this is not sufficiently nasty, you could always try the `discard` mailer.

Chapter 7

Oddities

7.1 Forwarding

- `ForwardPath`
- Normally just `$z/.forward`
- Consider `/var/mail/forwards/$z`
- Checks for stale NFS

Notes. . .

The default (and expected) behaviour is that *sendmail* should look for a file called `.forward` in the recipient's home directory. If it is found, and contains an email address (or a sequence of comma separated email addresses), the message is forwarded to those addresses. They can either be local to this computer, or can be addresses in any of the email messaging systems *sendmail* understands.

This can be used nicely for users who are away on customer sites (or perhaps just overseas) for extended periods of time. No system administrator time is involved in maintaining.

A nice option is *usermin* (`www.usermin.com`). This entirely web-based system lets users modify their `.forward` files in a moderately intuitive way.

Note that it is quite possible to make nasty mail loops with `.forward` files. For example, a user can send mail from A to B and back again. Or even create a `.forward` file on their mail server asking it to be delivered to their email address on that same mail server.

7.2 Non-standard delivery

`/etc/mail/aliases` and `.forward` can contain:

- \user
- |program
- `/some/file`
- :include:/some/file
- Local addresses
- Remote addresses

Notes. . .

Sometimes one address needs to be redirected and delivered normally. e.g. a tele-worker might want email delivered to their work email address and also to their ISP. Obviously if they edit their `.forward` file they can redirect it elsewhere, but if they want it also delivered locally, they can't just use "myname" in `.forward` – this would loop. The \user solves this problem by telling *sendmail* to skip any more `.forward` or `/etc/mail/aliases` lookups.

The program specified in |program should be given as a full path name. It will get run as *root* if it is in `aliases`, or as the user if it is in a `.forward`. The message is presented on standard input. If the program completes unsuccessfully (exit status non-zero) the message delivery is considered to have failed.

If there is a / at the beginning of the alias, it is probably a file. If there is a / at the beginning and there is also "@" somewhere, then it is considered an X500 address.

The file listed in `:include:` is read; each line is processed as if it were in the aliases file. (Or `.forward` file.) It can contain programs, files, users, email addresses, etc.

7.3 Fun ideas

- `ForwardPath=$z/.forward.$w`
- `ForwardPath=$z/.forward.$s`
- `somealias+extra: |program`
- `somealias+*: /else/where2`
- `owner-list: root`

Notes. . .

`$w` is the hostname of the computer *sendmail* is running on. So even if one's home directory is shared between several computers (e.g. it is on a USB keyring drive and you plug it in at home or at work), it is possible to differentiate the `.forward` files on each machine. This is rather odd, however!

`$r` is the sender of the message. Thus you can have separate `.forward` files depending on who sent you the message. This isn't particularly practical however – a better option is to use `procmail` to solve this kind of thing.

The third and fourth examples demonstrate what *sendmail* does with an email address containing a "+" sign. If an email comes to an address `somealias+extra`, naturally, it would get turned into the right hand side. The interesting case is an email to `somealias+other` – *sendmail* can't find an exact match, but can find a wild-card match with the `somealias+*`. If `somealias+*` were not there, *sendmail* would try looking for an alias just called "`somealias`" (i.e. without the plus sign at all), or a user called `somealias`.

Finally, if there is an error sending to the email address `list`, then it will be sent to *root* – *sendmail* looks for an address called "owner-XXX" for errors from XXX.

7.4 Vacation

- Program for auto-responding to emails
- Run from `.forward`
- Looks for a file (with headers) called `.vacation.msg`
- Will reply only once per address per week
- Keeps track in `.vacation.db`

Notes. . .

First you initialise the database

```
vacation -i
```

Then you write your `.vacation.msg` file, possibly including the string $SUBJECT:

> From: gregb@ifost.org.au
> Subject: Re: $SUBJECT
> Precedence: Bulk
>
> I am away on holidays this week. I will get your message when I get back, but I won't read it until then.
>
> - Greg

Then put this into `.forward`: `\gregb, "|/usr/bin/vacation -a greg.baker gregb"`

I still want it delivered into my mailbox – so I keep the \gregb. But I also want a reply sent.

Note the arguments given: no message will be sent unless the username (or `-a` alias) appears in the `To:` or `Cc:` headers. This helps stop vacation messages being sent back to a mailing list. Moreover, the precedence on mailing lists is supposed to be "bulk" or "list", and `vacation` will not respond to those messages either.

7.5 Exercises

Mailing lists and missing users

Notes. . .

1. Make a very tiny mailing list for a few users by creating an alias "staff" which reads from a list of staff members. Normally one would use genuine mailing-list software for this, such as majordomo (`www.greatcircle.com/majordomo`).
2. Create a user and send them away on vacation. Test it by mailing from another account.

Chapter 8

Header rewriting

8.1 Why modify headers?

- Because RFC822 demands it
- To include disclaimers
- To flag possible spam
- To reject messages

Notes...

RFC822 defines the following headers as *mandatory* – a mail transfer agent must make sure all of these are in place (and the mail user agent should make sure they are there in the first place):

- Received
- Date
- From
- Message-Id
- Resent-Date
- Resent-From
- Resent-Message-ID

sendmail can do other clever things with headers too, though. For example, we can run a header into a rule set to check that the "From:" field matches the MAIL FROM: given during the SMTP transaction.

8.2 A simple header addition

```
HX-Our-Extra-Stuff:  Flumph gloop
HX-Long-Stuff:  Garble warble
tab farble
```

Notes. . .

Placed in `sendmail.cf`, this will make sure that every mail message will have headers called “X-Our-Extra-Stuff” and “X-Long-Stuff”. Notice that the value for “X-Long-Stuff” wraps around on to a new line. The TAB and the newline character will be preserved into the header.

Headers are case-insensitive.

If you are playing around with headers, it is a good idea to use headers with names beginning with “X-” as RFC822 requests that mail processing software cope with all such headers. (They will generally leave them intact). You can’t just make up other headers; they will often get stripped off or modified; or sometimes, crash the mail handling program.

8.3 Exercise

Add in your own header. . .

Notes. . .

1. Add a header "X-Administrator" to all outgoing messages with your name in it, in case there is a mail problem and someone wants to get in touch with you.

8.4 Headers and macros

- HX-Size: `${msg_size}`
- HX-Received-Using: `$?rProto $r$.`

Notes. . .

We have added a (useless) header called “X-Size” which is set to the value of the macro `${msg_size}`. This is defined by *sendmail* when it receives a message.

`$r` is the protocol used to receive the message. This will usually be something like “ESMTP”. By putting the “`$?r`” in *sendmail*will fill it in the second half “Proto `$r`” only if `$r` exists.

There is also a kind of if-then-else using `$|`.

8.5 Conditional headers

- `H?x?Full-Name:  $x`
- Is **x** in the *flags* of the mailer handling this message?

Notes. . .

You might want to check back to page 58 for the possible flags that *sendmail* understands. You will notice that there are quite a few that are marked as “traditionally used by H rules”. Here is the list of them:

D This mailer wants a “Date:” header.

F This mailer wants a “From:” header.

M This mailer wants a “Message-Id:” header.

P This mailer wants a “Return-Path:” header.

x This mailer wants a “Full-Name:” header.

Now let’s look at the header definitions from a typical `sendmail.cf`:

```
H?P?Return-Path:  <$g>
HReceived:  $?sfrom $s $.$?_($?s$|from $.$_) $.
tab  $.by $j ($v/$Z)$?r with $r$.  id $i
tab  $?u for $u; $|; $.$b
H?D?Resent-Date:  $a
H?D?Date:  $a
H?F?Resent-From:  $?x$x <$g>$|$g$.
H?F?From:  $?x$x <$g>$|$g$.
H?x?Full-Name:  $x
H?M?Resent-Message-Id:  <$t.$i@$j>
H?M?Message-Id:  <$t.$i@$j>
```

$_ contains the `identd` identity of the service on the connecting host.

Note that most are optional, depending on what the mailer is expecting.

The only complicated one above is the “Received” header which has numerous nested $? constructs.

8.6 Exercise

Flags and headers

Notes. . .

1. Does the `local` mailer have the "l" flag set?
2. Are any headers conditional on "l"? Any commented-out headers? Enable them if they are commented out. Test to see that it works.
3. Add a simple header that will show $_ if it is set, or "unidentified" otherwise.

8.7 Complaining about headers

```
HMessage-Id:  $>CheckMsgId
...
SCheckMsgId
R< $+ @ $+ > tab $@ OK
R$* tab $#error $:  553 Header Err
```

Notes. . .

The ruleset `CheckMsgId` gets called on every message; it is given the value of the Message-Id field as its workspace. The ruleset on the overhead will let pass any message ID of the form "< *something* @ *something* >", and complain with anything else. The message will then get bounced.

A more lateral use of this functionality is to put a special header in for all the systems on your network that you do trust, and then to send any message that doesn't have your special header into the $#mailerdiscard mailer.

Incidentally, any comments (text in parentheses) in the value of the header will be removed before being passed to this ruleset. If you really wanted it preserved, just add a "+" symbol, thus:

```
HMessage-Id:  $>+CheckMsgId
```

8.8 Exercise

Enabling simple censorship…

Notes…

1. Reject all messages that have the word “sex” in their subject lines[1].

[1] Or, alternatively, if you run an adults-only site, perhaps you should reject anything that doesn’t.

Chapter 9

Simplifying everything

9.1 Don't edit `sendmail.cf`!

- Find your `.mc` files
- Change it
- Rerun `m4`
- Restart/HUP *sendmail*

Notes...

The format of the `.mc` is much, much nicer than the raw `sendmail.cf`.

There are several options for rerunning `m4`.

- `make install-cf` in the `cf/cf` directory
- Run `Build` from the `cf` directory.
- Run `m4 ../m4/cf.m4 myfile.mc`

We will use the final method as this is what the other ones run behind the scenes anyway.

Your `.mc` files should be in a sub directory `cf/cf` of the *sendmail* distribution. e.g. on OpenBSD, look in:

`/usr/src/gnu/usr.sbin/sendmail/cf/cf`

You will find a `README` file there, as well as many sample `.mc` files.

9.2 A simple `.mc` file

```
VERSIONID(‘Client -- sends mail elsewhere’)
OSTYPE(openbsd)
FEATURE(‘nullclient’,‘mailhub.ifost.org’)
```

Notes...

That’s a bit easier to develop than editing `sendmail.cf` directly!

9.3 Exercise

Autogenerating .cf files . . .

Notes. . .

1. Create the `.mc` file on page 86, with modifications that your instructor suggests.
2. Run `m4` as discussed on page 85. Send the output into `/etc/mail/myconf.cf`
3. Restart *sendmail* working of that new config file (`-C` option).
4. Send a mail message; where does it end up?

9.4 Things in a `.mc` file

VERSIONID Turns into a comment in `sendmail.cf`

OSTYPE Where files are found

FEATURE Turn on something

define Set a configuration option

dnl Delete to end of newline (comment)

MASQUERADE_AS

Notes. . .

There are a few other, rarer options:

DOMAIN Look in the `domain` directory for files that are common across our organisation

local macro definitions

MAILER Any extra mailers you want that can't be generated from features

LOCAL_CONFIG Configuration options you want set directly.

LOCAL_RULE_* Any modifications you want to make to local rulesets

LOCAL_RULESETS Any extra rulesets you want

We will look each of these options in turn.

9.5 VERSIONID

- Usually `$RCS: rcs id$`
- Can be anything
- Becomes a comment
- Keep in quotes ‘ and ’

Notes. . .

It’s a good idea to keep your `.mc` files under revision control. RCS (the simplest and easiest version control system to use) will update any string between $ signs.

`m4` will ignore anything between forward and backward quotes (i.e. start and end single quotes). Note this is *not* like the shell (a pair of forward single quotes) or most programming languages.

`m4` also will do nasty things to most text strings that include any `m4` keywords, so quoting things for protection is always a good idea.

9.6 OSTYPE

- Essential
- Defines where files go
- Not all operating systems defined
- Look in `ostype` for complete list

Notes. . .

From `cf/README`:

```
You MUST define an operating system environment, or the configuration
file build will puke.  There are several environments available; look
at the "ostype" directory for the current list.  This macro changes
things like the location of the alias file and queue directory.  Some
of these files are identical to one another.

It is IMPERATIVE that the OSTYPE occur before any MAILER definitions.
In general, the OSTYPE macro should go immediately after any version
information, and MAILER definitions should always go last.

Operating system definitions are usually easy to write.  They may define
the following variables (everything defaults, so an ostype file may be
empty).  Unfortunately, the list of configuration-supported systems is
not as broad as the list of source-supported systems, since many of
the source contributors do not include corresponding ostype files.

ALIAS_FILE              [/etc/mail/aliases] The location of the text version
                        of the alias file(s).  It can be a comma-separated
                        list of names (but be sure you quote values with
                        commas in them -- for example, use
                                define(‘ALIAS_FILE’, ‘a,b’)
                        to get "a" and "b" both listed as alias files;
                        otherwise the define() primitive only sees "a").
HELP_FILE               [/etc/mail/helpfile] The name of the file
                        containing information printed in response to
                        the SMTP HELP command.
QUEUE_DIR               [/var/spool/mqueue] The directory containing
                        queue files.  To use multiple queues, supply
                        a value ending with an asterisk.  For
                        example, /var/spool/mqueue/qd* will use all of the
                        directories or symbolic links to directories
                        beginning with ’qd’ in /var/spool/mqueue as queue
                        directories.  The names ’qf’, ’df’, and ’xf’ are
                        reserved as specific subdirectories for the
                        corresponding queue file types as explained in
                        doc/op/op.me.  See also QUEUE GROUP DEFINITIONS.
MSP_QUEUE_DIR           [/var/spool/clientmqueue] The directory containing
                        queue files for the MSP (Mail Submission Program,
                        see sendmail/SECURITY).
STATUS_FILE             [/etc/mail/statistics] The file containing status
                        information.
LOCAL_MAILER_PATH       [/bin/mail] The program used to deliver local mail.
LOCAL_MAILER_FLAGS      [Prmn9] The flags used by the local mailer.  The
                        flags lsDFMAw5:/|@q are always included.
LOCAL_MAILER_ARGS       [mail -d $u] The arguments passed to deliver local
                        mail.
LOCAL_MAILER_MAX        [undefined] If defined, the maximum size of local
                        mail that you are willing to accept.
LOCAL_MAILER_MAXMSGS    [undefined] If defined, the maximum number of
                        messages to deliver in a single connection.  Only
                        useful for LMTP local mailers.
LOCAL_MAILER_CHARSET    [undefined] If defined, messages containing 8-bit data
                        that ARRIVE from an address that resolves to the
                        local mailer and which are converted to MIME will be
                        labeled with this character set.
LOCAL_MAILER_EOL        [undefined] If defined, the string to use as the
                        end of line for the local mailer.
LOCAL_MAILER_DSN_DIAGNOSTIC_CODE
                        [X-Unix] The DSN Diagnostic-Code value for the
                        local mailer.  This should be changed with care.
LOCAL_SHELL_PATH        [/bin/sh] The shell used to deliver piped email.
LOCAL_SHELL_FLAGS       [eu9] The flags used by the shell mailer.  The
                        flags lsDFM are always included.
LOCAL_SHELL_ARGS        [sh -c $u] The arguments passed to deliver "prog"
                        mail.
LOCAL_SHELL_DIR         [$z:/] The directory search path in which the
                        shell should run.
LOCAL_MAILER_QGRP       [undefined] The queue group for the local mailer.
USENET_MAILER_PATH      [/usr/lib/news/inews] The name of the program
                        used to submit news.
USENET_MAILER_FLAGS     [rsDFMmn] The mailer flags for the usenet mailer.
USENET_MAILER_ARGS      [-m -h -n] The command line arguments for the
                        usenet mailer.  NOTE: Some versions of inews
                        (such as those shipped with newer versions of INN)
                        use different flags.  Double check the defaults
                        against the inews man page.
USENET_MAILER_MAX       [100000] The maximum size of messages that will
                        be accepted by the usenet mailer.
USENET_MAILER_QGRP      [undefined] The queue group for the usenet mailer.
SMTP_MAILER_FLAGS       [undefined] Flags added to SMTP mailer.  Default
                        flags are `mDFMuX' for all SMTP-based mailers; the
                        "esmtp" mailer adds `a'; "smtp8" adds `8'; and
                        "dsmtp" adds `%'.
RELAY_MAILER_FLAGS      [undefined] Flags added to the relay mailer.  Default
```

```
                        flags are `mDFMuX' for all SMTP-based mailers; the
                        relay mailer adds `a8'.  If this is not defined,
                        then SMTP_MAILER_FLAGS is used.
SMTP_MAILER_MAX         [undefined] The maximum size of messages that will
                        be transported using the smtp, smtp8, esmtp, or dsmtp
                        mailers.
SMTP_MAILER_MAXMSGS     [undefined] If defined, the maximum number of
                        messages to deliver in a single connection for the
                        smtp, smtp8, esmtp, or dsmtp mailers.
SMTP_MAILER_ARGS        [TCP $h] The arguments passed to the smtp mailer.
                        About the only reason you would want to change this
                        would be to change the default port.
ESMTP_MAILER_ARGS       [TCP $h] The arguments passed to the esmtp mailer.
SMTP8_MAILER_ARGS       [TCP $h] The arguments passed to the smtp8 mailer.
DSMTP_MAILER_ARGS       [TCP $h] The arguments passed to the dsmtp mailer.
RELAY_MAILER_ARGS       [TCP $h] The arguments passed to the relay mailer.
SMTP_MAILER_QGRP        [undefined] The queue group for the smtp mailer.
ESMTP_MAILER_QGRP       [undefined] The queue group for the esmtp mailer.
SMTP8_MAILER_QGRP       [undefined] The queue group for the smtp8 mailer.
DSMTP_MAILER_QGRP       [undefined] The queue group for the dsmtp mailer.
RELAY_MAILER_QGRP       [undefined] The queue group for the relay mailer.
RELAY_MAILER_MAXMSGS    [undefined] If defined, the maximum number of
                        messages to deliver in a single connection for the
                        relay mailer.
SMTP_MAILER_CHARSET     [undefined] If defined, messages containing 8-bit data
                        that ARRIVE from an address that resolves to one of
                        the SMTP mailers and which are converted to MIME will
                        be labeled with this character set.
UUCP_MAILER_PATH        [/usr/bin/uux] The program used to send UUCP mail.
UUCP_MAILER_FLAGS       [undefined] Flags added to UUCP mailer.  Default
                        flags are `DFMhuU' (and `m' for uucp-new mailer,
                        minus `U' for uucp-dom mailer).
UUCP_MAILER_ARGS        [uux - -r -z -a$g -gC $h!rmail ($u)] The arguments
                        passed to the UUCP mailer.
UUCP_MAILER_MAX         [100000] The maximum size message accepted for
                        transmission by the UUCP mailers.
UUCP_MAILER_CHARSET     [undefined] If defined, messages containing 8-bit data
                        that ARRIVE from an address that resolves to one of
                        the UUCP mailers and which are converted to MIME will
                        be labeled with this character set.
UUCP_MAILER_QGRP        [undefined] The queue group for the UUCP mailers.
FAX_MAILER_PATH         [/usr/local/lib/fax/mailfax] The program used to
                        submit FAX messages.
FAX_MAILER_ARGS         [mailfax $u $h $f] The arguments passed to the FAX
                        mailer.
FAX_MAILER_MAX          [100000] The maximum size message accepted for
                        transmission by FAX.
POP_MAILER_PATH         [/usr/lib/mh/spop] The pathname of the POP mailer.
POP_MAILER_FLAGS        [Penu] Flags added to POP mailer.  Flags lsDFMq
                        are always added.
```

```
POP_MAILER_ARGS          [pop $u] The arguments passed to the POP mailer.
POP_MAILER_QGRP          [undefined] The queue group for the pop mailer.
PROCMAIL_MAILER_PATH     [/usr/local/bin/procmail] The path to the procmail
                         program.  This is also used by
                         FEATURE(`local_procmail').
PROCMAIL_MAILER_FLAGS    [SPhnu9] Flags added to Procmail mailer.  Flags
                         DFM are always set.  This is NOT used by
                         FEATURE(`local_procmail'); tweak LOCAL_MAILER_FLAGS
                         instead.
PROCMAIL_MAILER_ARGS     [procmail -Y -m $h $f $u] The arguments passed to
                         the Procmail mailer.  This is NOT used by
                         FEATURE(`local_procmail'); tweak LOCAL_MAILER_ARGS
                         instead.
PROCMAIL_MAILER_MAX      [undefined] If set, the maximum size message that
                         will be accepted by the procmail mailer.
PROCMAIL_MAILER_QGRP     [undefined] The queue group for the procmail mailer.
MAIL11_MAILER_PATH       [/usr/etc/mail11] The path to the mail11 mailer.
MAIL11_MAILER_FLAGS      [nsFx] Flags for the mail11 mailer.
MAIL11_MAILER_ARGS       [mail11 $g $x $h $u] Arguments passed to the mail11
                         mailer.
MAIL11_MAILER_QGRP       [undefined] The queue group for the mail11 mailer.
PH_MAILER_PATH           [/usr/local/etc/phquery] The path to the phquery
                         program.
PH_MAILER_FLAGS          [ehmu] Flags for the phquery mailer.  Flags nrDFM
                         are always set.
PH_MAILER_ARGS           [phquery -- $u] -- arguments to the phquery mailer.
PH_MAILER_QGRP           [undefined] The queue group for the ph mailer.
CYRUS_MAILER_FLAGS       [Ah5@/:|] The flags used by the cyrus mailer.  The
                         flags lsDFMnPq are always included.
CYRUS_MAILER_PATH        [/usr/cyrus/bin/deliver] The program used to deliver
                         cyrus mail.
CYRUS_MAILER_ARGS        [deliver -e -m $h -- $u] The arguments passed
                         to deliver cyrus mail.
CYRUS_MAILER_MAX         [undefined] If set, the maximum size message that
                         will be accepted by the cyrus mailer.
CYRUS_MAILER_USER        [cyrus:mail] The user and group to become when
                         running the cyrus mailer.
CYRUS_MAILER_QGRP        [undefined] The queue group for the cyrus mailer.
CYRUS_BB_MAILER_FLAGS    [u] The flags used by the cyrusbb mailer.
                         The flags lsDFMnP are always included.
CYRUS_BB_MAILER_ARGS     [deliver -e -m $u] The arguments passed
                         to deliver cyrusbb mail.
confEBINDIR              [/usr/libexec] The directory for executables.
                         Currently used for FEATURE(`local_lmtp') and
                         FEATURE(`smrsh').
QPAGE_MAILER_FLAGS       [mDFMs] The flags used by the qpage mailer.
QPAGE_MAILER_PATH        [/usr/local/bin/qpage] The program used to deliver
                         qpage mail.
QPAGE_MAILER_ARGS        [qpage -l0 -m -P$u] The arguments passed
                         to deliver qpage mail.
```

```
QPAGE_MAILER_MAX        [4096] If set, the maximum size message that
                        will be accepted by the qpage mailer.
QPAGE_MAILER_QGRP       [undefined] The queue group for the qpage mailer.
LOCAL_PROG_QGRP         [undefined] The queue group for the prog mailer.
```

Note: to tweak Name_MAILER_FLAGS use the macro MODIFY_MAILER_FLAGS: MODIFY_MAILER_FLAGS(‘Name’, ‘change’) where Name is the first part of the macro Name_MAILER_FLAGS and change can be: flags that should be used directly (thus overriding the default value), or if it starts with ‘+’ (‘-’) then those flags are added to (removed from) the default value. Example:

```
        MODIFY_MAILER_FLAGS(‘LOCAL’, ‘+e’)
```

will add the flag ‘e’ to LOCAL_MAILER_FLAGS. Notice: there are several smtp mailers all of which are manipulated individually. See the section MAILERS for the available mailer names.
WARNING: The FEATUREs local_lmtp and local_procmail set LOCAL_MAILER_FLAGS unconditionally, i.e., without respecting any definitions in an OSTYPE setting.

9.7 Famous FEATURES

use_cw_file Read an /etc/mail/local-host-names

redirect Control users who have moved

virtusertable Handle virtual domains

local_procmail Use procmail as a local mailer

dnsbl Stop known spammers

Notes. . .

From the cf/README:

A FEATURE may contain up to 9 optional parameters – for example:

```
FEATURE(‘mailertable’, ‘dbm /usr/lib/mailertable’)
```

The default database map type for the table features can be set with

```
define(‘DATABASE_MAP_TYPE’, ‘dbm’)
```

which would set it to use ndbm databases. The default is the Berkeley DB hash database format. Note that you must still declare a database map type if you specify an argument to a FEATURE. DATABASE_MAP_TYPE is only used if no argument is given for the FEATURE. It must be specified before any feature that uses a map.

Also, features which can take a map definition as an argument can also take the special keyword ‘LDAP’. If that keyword is used, the map will use the LDAP definition described in the “USING LDAP FOR ALIASES, MAPS, AND CLASSES” section below.

Available features are:

```
use_cw_file     Read the file /etc/mail/local-host-names file to get
                alternate names for this host.  This might be used if you
                were on a host that MXed for a dynamic set of other hosts.
                If the set is static, just including the line "Cw<name1>
                <name2> ..." (where the names are fully qualified domain
                names) is probably superior.  The actual filename can be
                overridden by redefining confCW_FILE.

use_ct_file     Read the file /etc/mail/trusted-users file to get the
                names of users that will be ‘‘trusted’’, that is, able to
                set their envelope from address using -f without generating
                a warning message.  The actual filename can be overridden
                by redefining confCT_FILE.
```

```
redirect        Reject all mail addressed to "address.REDIRECT" with
                a ``551 User has moved; please try <address>'' message.
                If this is set, you can alias people who have left
                to their new address with ".REDIRECT" appended.

nouucp          Don't route UUCP addresses.  This feature takes one
                parameter:
                `reject': reject addresses which have "!" in the local
                        part unless it originates from a system
                        that is allowed to relay.
                `nospecial': don't do anything special with "!".
                Warnings: 1. See the notice in the anti-spam section.
                2. don't remove "!" from OperatorChars if `reject' is
                given as parameter.

nocanonify      Don't pass addresses to $[ ... $] for canonification
                by default, i.e., host/domain names are considered canonical,
                except for unqualified names, which must not be used in this
                mode (violation of the standard).  It can be changed by
                setting the DaemonPortOptions modifiers (M=).  That is,
                FEATURE(`nocanonify') will be overridden by setting the
                'c' flag.  Conversely, if FEATURE(`nocanonify') is not used,
                it can be emulated by setting the 'C' flag
                (DaemonPortOptions=Modifiers=C).  This would generally only
                be used by sites that only act as mail gateways or which have
                user agents that do full canonification themselves.  You may
                also want to use
                "define(`confBIND_OPTS', `-DNSRCH -DEFNAMES')" to turn off
                the usual resolver options that do a similar thing.

                An exception list for FEATURE(`nocanonify') can be
                specified with CANONIFY_DOMAIN or CANONIFY_DOMAIN_FILE,
                i.e., a list of domains which are nevertheless passed to
                $[ ... $] for canonification.  This is useful to turn on
                canonification for local domains, e.g., use
                CANONIFY_DOMAIN(`my.domain my') to canonify addresses
                which end in "my.domain" or "my".
                Another way to require canonification in the local
                domain is CANONIFY_DOMAIN(`$=m').

                A trailing dot is added to addresses with more than
                one component in it such that other features which
                expect a trailing dot (e.g., virtusertable) will
                still work.

                If `canonify_hosts' is specified as parameter, i.e.,
                FEATURE(`nocanonify', `canonify_hosts'), then
                addresses which have only a hostname, e.g.,
                <user@host>, will be canonified (and hopefully fully
                qualified), too.

stickyhost      This feature is sometimes used with LOCAL_RELAY,
                although it can be used for a different effect with
                MAIL_HUB.

                When used without MAIL_HUB, email sent to
                "user@local.host" are marked as "sticky" -- that
                is, the local addresses aren't matched against UDB,
                don't go through ruleset 5, and are not forwarded to
                the LOCAL_RELAY (if defined).

                With MAIL_HUB, mail addressed to "user@local.host"
                is forwarded to the mail hub, with the envelope
                address still remaining "user@local.host".
                Without stickyhost, the envelope would be changed
                to "user@mail_hub", in order to protect against
                mailing loops.

mailertable     Include a "mailer table" which can be used to override
                routing for particular domains (which are not in class {w},
                i.e.  local host names).  The argument of the FEATURE may be
                the key definition.  If none is specified, the definition
                used is:

                        hash /etc/mail/mailertable

                Keys in this database are fully qualified domain names
                or partial domains preceded by a dot -- for example,
                "vangogh.CS.Berkeley.EDU" or ".CS.Berkeley.EDU".  As a
                special case of the latter, "." matches any domain not
                covered by other keys.  Values must be of the form:
                        mailer:domain
                where "mailer" is the internal mailer name, and "domain"
                is where to send the message.  These maps are not
                reflected into the message header.  As a special case,
                the forms:
                        local:user
                will forward to the indicated user using the local mailer,
                        local:
                will forward to the original user in the e-mail address
                using the local mailer, and
                        error:code message
                        error:D.S.N:code message
                will give an error message with the indicated SMTP reply
                code and message, where D.S.N is an RFC 1893 compliant
                error code.

domaintable     Include a "domain table" which can be used to provide
                domain name mapping.  Use of this should really be
```

limited to your own domains. It may be useful if you change names (e.g., your company changes names from oldname.com to newname.com). The argument of the FEATURE may be the key definition. If none is specified, the definition used is:

```
hash /etc/mail/domaintable
```

The key in this table is the domain name; the value is the new (fully qualified) domain. Anything in the domaintable is reflected into headers; that is, this is done in ruleset 3.

bitdomain

Look up bitnet hosts in a table to try to turn them into internet addresses. The table can be built using the bitdomain program contributed by John Gardiner Myers. The argument of the FEATURE may be the key definition; if none is specified, the definition used is:

```
hash /etc/mail/bitdomain
```

Keys are the bitnet hostname; values are the corresponding internet hostname.

uucpdomain

Similar feature for UUCP hosts. The default map definition is:

```
hash /etc/mail/uudomain
```

At the moment there is no automagic tool to build this database.

always_add_domain

Include the local host domain even on locally delivered mail. Normally it is not added on unqualified names. However, if you use a shared message store but do not use the same user name space everywhere, you may need the host name on local names. An optional argument specifies another domain to be added than the local.

allmasquerade

If masquerading is enabled (using MASQUERADE_AS), this feature will cause recipient addresses to also masquerade as being from the masquerade host. Normally they get the local hostname. Although this may be right for ordinary users, it can break local aliases. For example, if you send to "localalias", the originating sendmail will find that alias and send to all members, but send the message with "To: localalias@masqueradehost". Since that alias likely does not exist, replies will fail. Use this feature ONLY if you can guarantee that the ENTIRE namespace on your masquerade host supersets all the local entries.

limited_masquerade

Normally, any hosts listed in class {w} are masqueraded. If this feature is given, only the hosts listed in class {M} (see below: MASQUERADE_DOMAIN) are masqueraded. This is useful if you have several domains with disjoint namespaces hosted on the same machine.

masquerade_entire_domain

If masquerading is enabled (using MASQUERADE_AS) and MASQUERADE_DOMAIN (see below) is set, this feature will cause addresses to be rewritten such that the masquerading domains are actually entire domains to be hidden. All hosts within the masquerading domains will be rewritten to the masquerade name (used in MASQUERADE_AS). For example, if you have:

```
MASQUERADE_AS(‘masq.com’)
MASQUERADE_DOMAIN(‘foo.org’)
MASQUERADE_DOMAIN(‘bar.com’)
```

then *foo.org and *bar.com are converted to masq.com. Without this feature, only foo.org and bar.com are masqueraded.

> NOTE: only domains within your jurisdiction and current hierarchy should be masqueraded using this.

local_no_masquerade

This feature prevents the local mailer from masquerading even if MASQUERADE_AS is used. MASQUERADE_AS will only have effect on addresses of mail going outside the local domain.

genericstable

This feature will cause unqualified addresses (i.e., without a domain) and addresses with a domain listed in class {G} to be looked up in a map and turned into another ("generic") form, which can change both the domain name and the user name. Notice: if you use an MSP (as it is default starting with 8.12), the MTA will only receive qualified addresses from the MSP (as required by the RFCs). Hence you need to add your domain to class {G}. This feature is similar to the userdb functionality. The same types of addresses as for masquerading are looked up, i.e., only header sender addresses unless the allmasquerade and/or masquerade_envelope features are given. Qualified addresses must have the domain part in class {G}; entries can be added to this class by the macros GENERICS_DOMAIN or GENERICS_DOMAIN_FILE (analogously to MASQUERADE_DOMAIN and MASQUERADE_DOMAIN_FILE, see below).

The argument of FEATURE(‘genericstable’) may be the map definition; the default map definition is:

```
hash /etc/mail/genericstable
```

The key for this table is either the full address, the domain (with a leading @; the localpart is passed as first argument) or the unqualified username (tried in the order mentioned); the value is the new user address. If the new user address does not include a domain, it will be qualified in the standard manner, i.e., using $j or the masquerade name. Note that the address being looked up must be fully qualified. For local mail, it is necessary to use FEATURE(‘always_add_domain’) for the addresses to be qualified.
The "+detail" of an address is passed as %1, so entries like

```
old+*@foo.org   new+%1@example.com
gen+*@foo.org   %1@example.com
```

and other forms are possible.

generics_entire_domain
If the genericstable is enabled and GENERICS_DOMAIN or GENERICS_DOMAIN_FILE is used, this feature will cause addresses to be searched in the map if their domain parts are subdomains of elements in class {G}.

virtusertable A domain-specific form of aliasing, allowing multiple virtual domains to be hosted on one machine. For example, if the virtuser table contained:

```
info@foo.com    foo-info
info@bar.com    bar-info
joe@bar.com     error:nouser 550 No such user here
jax@bar.com     error:5.7.0:550 Address invalid
@baz.org        jane@example.net
```

then mail addressed to info@foo.com will be sent to the address foo-info, mail addressed to info@bar.com will be delivered to bar-info, and mail addressed to anyone at baz.org will be sent to jane@example.net, mail to joe@bar.com will be rejected with the specified error message, and mail to jax@bar.com will also have a RFC 1893 compliant error code 5.7.0.

The username from the original address is passed as %1 allowing:

```
@foo.org        %1@example.com
```

meaning someone@foo.org will be sent to someone@example.com. Additionally, if the local part consists of "user+detail" then "detail" is passed as %2 and "+detail" is passed as %3 when a match against user+* is attempted, so entries like

```
        old+*@foo.org   new+%2@example.com
        gen+*@foo.org   %2@example.com
        +*@foo.org      %1%3@example.com
        X++@foo.org     Z%3@example.com
        @bar.org        %1%3
```

and other forms are possible. Note: to preserve "+detail" for a default case (@domain) %1%3 must be used as RHS. There are two wildcards after "+": "+" matches only a non-empty detail, "*" matches also empty details, e.g., user+@foo.org matches +*@foo.org but not ++@foo.org. This can be used to ensure that the parameters %2 and %3 are not empty.

All the host names on the left hand side (foo.com, bar.com, and baz.org) must be in class {w} or class {VirtHost}. The latter can be defined by the macros VIRTUSER_DOMAIN or VIRTUSER_DOMAIN_FILE (analogously to MASQUERADE_DOMAIN and MASQUERADE_DOMAIN_FILE, see below). If VIRTUSER_DOMAIN or VIRTUSER_DOMAIN_FILE is used, then the entries of class {VirtHost} are added to class {R}, i.e., relaying is allowed to (and from) those domains. The default map definition is:

```
        hash /etc/mail/virtusertable
```

A new definition can be specified as the second argument of the FEATURE macro, such as

```
        FEATURE(‘virtusertable’, ‘dbm /etc/mail/virtusers’)
```

virtuser_entire_domain

If the virtusertable is enabled and VIRTUSER_DOMAIN or VIRTUSER_DOMAIN_FILE is used, this feature will cause addresses to be searched in the map if their domain parts are subdomains of elements in class {VirtHost}.

ldap_routing

Implement LDAP-based e-mail recipient routing according to the Internet Draft draft-lachman-laser-ldap-mail-routing-01. This provides a method to re-route addresses with a domain portion in class {LDAPRoute} to either a different mail host or a different address. Hosts can be added to this class using LDAPROUTE_DOMAIN and LDAPROUTE_DOMAIN_FILE (analogously to MASQUERADE_DOMAIN and MASQUERADE_DOMAIN_FILE, see below).

See the LDAP ROUTING section below for more information.

```
nodns           If you aren't running DNS at your site (for example,
                you are UUCP-only connected).  It's hard to consider
                this a "feature", but hey, it had to go somewhere.
                Actually, as of 8.7 this is a no-op -- remove "dns" from
                the hosts service switch entry instead.

nullclient      This is a special case -- it creates a configuration file
                containing nothing but support for forwarding all mail to a
                central hub via a local SMTP-based network.  The argument
                is the name of that hub.

                The only other feature that should be used in conjunction
                with this one is FEATURE(`nocanonify').  No mailers
                should be defined.  No aliasing or forwarding is done.

local_lmtp      Use an LMTP capable local mailer.  The argument to this
                feature is the pathname of an LMTP capable mailer.  By
                default, mail.local is used.  This is expected to be the
                mail.local which came with the 8.9 distribution which is
                LMTP capable.  The path to mail.local is set by the
                confEBINDIR m4 variable -- making the default
                LOCAL_MAILER_PATH /usr/libexec/mail.local.
                WARNING: This feature sets LOCAL_MAILER_FLAGS unconditionally,
                i.e., without respecting any definitions in an OSTYPE setting.

local_procmail  Use procmail or another delivery agent as the local mailer.
                The argument to this feature is the pathname of the
                delivery agent, which defaults to PROCMAIL_MAILER_PATH.
                Note that this does NOT use PROCMAIL_MAILER_FLAGS or
                PROCMAIL_MAILER_ARGS for the local mailer; tweak
                LOCAL_MAILER_FLAGS and LOCAL_MAILER_ARGS instead, or
                specify the appropriate parameters.  When procmail is used,
                the local mailer can make use of the
                "user+indicator@local.host" syntax; normally the +indicator
                is just tossed, but by default it is passed as the -a
                argument to procmail.

                This feature can take up to three arguments:

                1. Path to the mailer program
                   [default: /usr/local/bin/procmail]
                2. Argument vector including name of the program
                   [default: procmail -Y -a $h -d $u]
                3. Flags for the mailer [default: SPfhn9]

                Empty arguments cause the defaults to be taken.

                For example, this allows it to use the maildrop
                (http://www.flounder.net/~mrsam/maildrop/) mailer instead
                by specifying:

                FEATURE(`local_procmail', `/usr/local/bin/maildrop',
                 `maildrop -d $u')

                or scanmails using:

                FEATURE(`local_procmail', `/usr/local/bin/scanmails')

                WARNING: This feature sets LOCAL_MAILER_FLAGS unconditionally,
                i.e.,  without respecting any definitions in an OSTYPE setting.

bestmx_is_local Accept mail as though locally addressed for any host that
                lists us as the best possible MX record.  This generates
                additional DNS traffic, but should be OK for low to
                medium traffic hosts.  The argument may be a set of
                domains, which will limit the feature to only apply to
                these domains -- this will reduce unnecessary DNS
                traffic.  THIS FEATURE IS FUNDAMENTALLY INCOMPATIBLE WITH
                WILDCARD MX RECORDS!!!  If you have a wildcard MX record
                that matches your domain, you cannot use this feature.

smrsh           Use the SendMail Restricted SHell (smrsh) provided
                with the distribution instead of /bin/sh for mailing
                to programs.  This improves the ability of the local
                system administrator to control what gets run via
                e-mail.  If an argument is provided it is used as the
                pathname to smrsh; otherwise, the path defined by
                confEBINDIR is used for the smrsh binary -- by default,
                /usr/libexec/smrsh is assumed.

promiscuous_relay
                By default, the sendmail configuration files do not permit
                mail relaying (that is, accepting mail from outside your
                local host (class {w}) and sending it to another host than
                your local host).  This option sets your site to allow
                mail relaying from any site to any site.  In almost all
                cases, it is better to control relaying more carefully
                with the access map, class {R}, or authentication.  Domains
                can be added to class {R} by the macros RELAY_DOMAIN or
                RELAY_DOMAIN_FILE (analogously to MASQUERADE_DOMAIN and
                MASQUERADE_DOMAIN_FILE, see below).

relay_entire_domain
                By default, only hosts listed as RELAY in the access db
                will be allowed to relay.  This option also allows any
                host in your domain as defined by class {m}.

relay_hosts_only
                By default, names that are listed as RELAY in the access
```

db and class {R} are domain names, not host names. For example, if you specify ‘‘foo.com’’, then mail to or from foo.com, abc.foo.com, or a.very.deep.domain.foo.com will all be accepted for relaying. This feature changes the behaviour to lookup individual host names only.

relay_based_on_MX

Turns on the ability to allow relaying based on the MX records of the host portion of an incoming recipient; that is, if an MX record for host foo.com points to your site, you will accept and relay mail addressed to foo.com. See description below for more information before using this feature. Also, see the KNOWNBUGS entry regarding bestmx map lookups.

FEATURE(‘relay_based_on_MX’) does not necessarily allow routing of these messages which you expect to be allowed, if route address syntax (or %-hack syntax) is used. If this is a problem, add entries to the access-table or use FEATURE(‘loose_relay_check’).

relay_mail_from

Allows relaying if the mail sender is listed as RELAY in the access map. If an optional argument ‘domain’ is given, relaying can be allowed just based on the domain portion of the sender address. This feature should only be used if absolutely necessary as the sender address can be easily forged. Use of this feature requires the "From:" tag be prepended to the key in the access map; see the discussion of tags and FEATURE(‘relay_mail_from’) in the section on anti-spam configuration control.

relay_local_from

Allows relaying if the domain portion of the mail sender is a local host. This should only be used if absolutely necessary as it opens a window for spammers. Specifically, they can send mail to your mail server that claims to be from your domain (either directly or via a routed address), and you will go ahead and relay it out to arbitrary hosts on the Internet.

accept_unqualified_senders

Normally, MAIL FROM: commands in the SMTP session will be refused if the connection is a network connection and the sender address does not include a domain name. If your setup sends local mail unqualified (i.e., MAIL FROM: <joe>), you will need to use this feature to accept unqualified sender addresses. Setting the DaemonPortOptions modifier ’u’ overrides the default behavior, i.e., unqualified addresses are accepted even without this FEATURE.

```
                If this FEATURE is not used, the DaemonPortOptions modifier
                'f' can be used to enforce fully qualified addresses.

accept_unresolvable_domains
                Normally, MAIL FROM: commands in the SMTP session will be
                refused if the host part of the argument to MAIL FROM:
                cannot be located in the host name service (e.g., an A or
                MX record in DNS).  If you are inside a firewall that has
                only a limited view of the Internet host name space, this
                could cause problems.  In this case you probably want to
                use this feature to accept all domains on input, even if
                they are unresolvable.

access_db       Turns on the access database feature.  The access db gives
                you the ability to allow or refuse to accept mail from
                specified domains for administrative reasons.  Moreover,
                it can control the behavior of sendmail in various situations.
                By default, the access database specification is:

                        hash -T<TMPF> /etc/mail/access

                See the anti-spam configuration control section for further
                important information about this feature.  Notice:
                "-T<TMPF>" is meant literal, do not replace it by anything.

blacklist_recipients
                Turns on the ability to block incoming mail for certain
                recipient usernames, hostnames, or addresses.  For
                example, you can block incoming mail to user nobody,
                host foo.mydomain.com, or guest@bar.mydomain.com.
                These specifications are put in the access db as
                described in the anti-spam configuration control section
                later in this document.

delay_checks    The rulesets check_mail and check_relay will not be called
                when a client connects or issues a MAIL command, respectively.
                Instead, those rulesets will be called by the check_rcpt
                ruleset; they will be skipped under certain circumstances.
                See "Delay all checks" in the anti-spam configuration control
                section.  Note: this feature is incompatible to the versions
                in 8.10 and 8.11.

dnsbl           Turns on rejection of hosts found in an DNS based rejection
                list.  If an argument is provided it is used as the domain
                in which blocked hosts are listed; otherwise it defaults to
                blackholes.mail-abuse.org.  An explanation for an DNS based
                rejection list can be found at http://mail-abuse.org/rbl/.
                A second argument can be used to change the default error
                message.  Without that second argument, the error message
                will be
```

        Mail from IP-ADDRESS refused by blackhole site SERVER

where IP-ADDRESS and SERVER are replaced by the appropriate information. By default, temporary lookup failures are ignored. This behavior can be changed by specifying a third argument, which must be either ‘t’ or a full error message. See the anti-spam configuration control section for an example. The dnsbl feature can be included several times to query different DNS based rejection lists. See also enhdnsbl for an enhanced version.

NOTE: The default DNS blacklist, blackholes.mail-abuse.org, is a service offered by the Mail Abuse Prevention System (MAPS). As of July 31, 2001, MAPS is a subscription service, so using that network address won’t work if you haven’t subscribed. Contact MAPS to subscribe (http://mail-abuse.org/).

enhdnsbl

Enhanced version of dnsbl (see above). Further arguments (up to 5) can be used to specify specific return values from lookups. Temporary lookup failures are ignored unless a third argument is given, which must be either ‘t’ or a full error message. By default, any successful lookup will generate an error. Otherwise the result of the lookup is compared with the supplied argument(s), and only if a match occurs an error is generated. For example,

```
FEATURE(‘enhdnsbl’, ‘dnsbl.example.com’, ‘’, ‘t’, ‘127.0.0.2.’)
```

will reject the e-mail if the lookup returns the value ‘‘127.0.0.2.’’, or generate a 451 response if the lookup temporarily failed. The arguments can contain metasymbols as they are allowed in the LHS of rules. As the example shows, the default values are also used if an empty argument, i.e., ‘’, is specified. This feature requires that sendmail has been compiled with the flag DNSMAP (see sendmail/README).

lookupdotdomain

Look up also .domain in the access map. This allows to match only subdomains. It does not work well with FEATURE(‘relay_hosts_only’), because most lookups for subdomains are suppressed by the latter feature.

loose_relay_check

Normally, if % addressing is used for a recipient, e.g. user%site@othersite, and othersite is in class {R}, the check_rcpt ruleset will strip @othersite and recheck user@site for relaying. This feature changes that behavior. It should not be needed for most installations.

authinfo

Provide a separate map for client side authentication information. See SMTP AUTHENTICATION for details.

```
                By default, the authinfo database specification is:

                        hash /etc/mail/authinfo

preserve_luser_host
                Preserve the name of the recipient host if LUSER_RELAY is
                used.  Without this option, the domain part of the
                recipient address will be replaced by the host specified as
                LUSER_RELAY.  This feature only works if the hostname is
                passed to the mailer (see mailer triple in op.me).  Note
                that in the default configuration the local mailer does not
                receive the hostname, i.e., the mailer triple has an empty
                hostname.

preserve_local_plus_detail
                Preserve the +detail portion of the address when passing
                address to local delivery agent.  Disables alias and
                .forward +detail stripping (e.g., given user+detail, only
                that address will be looked up in the alias file; user+* and
                user will not be looked up).  Only use if the local
                delivery agent in use supports +detail addressing.

compat_check    Enable ruleset check_compat to look up pairs of addresses
                with the Compat: tag -- Compat:sender<@>recipient -- in the
                access map.  Valid values for the RHS include
                        DISCARD silently discard recipient
                        TEMP:   return a temporary error
                        ERROR:  return a permanent error
                In the last two cases, a 4xy/5xy SMTP reply code should
                follow the colon.

no_default_msa  Don’t generate the default MSA daemon, i.e.,
                DAEMON_OPTIONS(‘Port=587,Name=MSA,M=E’)
                To define a MSA daemon with other parameters, use this
                FEATURE and introduce new settings via DAEMON_OPTIONS().

msp             Defines config file for Message Submission Program.
                See sendmail/SECURITY for details and cf/cf/submit.mc
                how to use it.  An optional argument can be used to
                override the default of ‘localhost’ to use as host to send
                all e-mails to.  If ‘MSA’ is specified as second argument
                then port 587 is used to contact the server.  Example:

                        FEATURE(‘msp’, ‘’, ‘MSA’)

                Some more hints about possible changes can be found below
                in the section MESSAGE SUBMISSION PROGRAM.

queuegroup      A simple example how to select a queue group based
                on the full e-mail address or the domain of the
```

recipient. Selection is done via entries in the access map using the tag QGRP:, for example:

```
        QGRP:example.com        main
        QGRP:friend@some.org    others
        QGRP:my.domain          local
```

where "main", "others", and "local" are names of queue groups. If an argument is specified, it is used as default queue group.

9.8 MASQUERADING

- MASQUERADE_AS(‘company.com‘)
- MASQUERADE_DOMAIN(‘oldcompanyname.com‘)
- MASQUERADE_DOMAIN_FILE(‘filename‘)

Notes. . .

From cf/README:

```
You can have your host masquerade as another using

        MASQUERADE_AS(‘host.domain’)

This causes mail being sent to be labeled as coming from the
indicated host.domain, rather than $j.  One normally masquerades as
one of one’s own subdomains (for example, it’s unlikely that
Berkeley would choose to masquerade as an MIT site).  This
behaviour is modified by a plethora of FEATUREs; in particular, see
masquerade_envelope, allmasquerade, limited_masquerade, and
masquerade_entire_domain.

The masquerade name is not normally canonified, so it is important
that it be your One True Name, that is, fully qualified and not a
CNAME.  However, if you use a CNAME, the receiving side may canonify
it for you, so don’t think you can cheat CNAME mapping this way.

Normally the only addresses that are masqueraded are those that come
from this host (that is, are either unqualified or in class {w}, the list
of local domain names).  You can augment this list, which is realized
by class {M} using

        MASQUERADE_DOMAIN(‘otherhost.domain’)

The effect of this is that although mail to user@otherhost.domain
will not be delivered locally, any mail including any user@otherhost.domain
will, when relayed, be rewritten to have the MASQUERADE_AS address.
This can be a space-separated list of names.

If these names are in a file, you can use

        MASQUERADE_DOMAIN_FILE(‘filename’)

to read the list of names from the indicated file (i.e., to add
elements to class {M}).
```

To exempt hosts or subdomains from being masqueraded, you can use

```
        MASQUERADE_EXCEPTION(‘host.domain’)
```

This can come handy if you want to masquerade a whole domain except for one (or a few) host(s). If these names are in a file, you can use

```
        MASQUERADE_EXCEPTION_FILE(‘filename’)
```

Normally only header addresses are masqueraded. If you want to masquerade the envelope as well, use

```
        FEATURE(‘masquerade_envelope’)
```

There are always users that need to be "exposed" -- that is, their internal site name should be displayed instead of the masquerade name. Root is an example (which has been "exposed" by default prior to 8.10). You can add users to this list using

```
        EXPOSED_USER(‘usernames’)
```

This adds users to class {E}; you could also use

```
        EXPOSED_USER_FILE(‘filename’)
```

You can also arrange to relay all unqualified names (that is, names without @host) to a relay host. For example, if you have a central email server, you might relay to that host so that users don’t have to have .forward files or aliases. You can do this using

```
        define(‘LOCAL_RELAY’, ‘mailer:hostname’)
```

The ‘‘mailer:’’ can be omitted, in which case the mailer defaults to "relay". There are some user names that you don’t want relayed, perhaps because of local aliases. A common example is root, which may be locally aliased. You can add entries to this list using

```
        LOCAL_USER(‘usernames’)
```

This adds users to class {L}; you could also use

```
        LOCAL_USER_FILE(‘filename’)
```

If you want all incoming mail sent to a centralized hub, as for a shared /var/spool/mail scheme, use

```
        define(‘MAIL_HUB’, ‘mailer:hostname’)
```

Again, ‘‘mailer:’’ defaults to "relay". If you define both LOCAL_RELAY and MAIL_HUB _AND_ you have FEATURE(‘stickyhost’), unqualified names will be sent to the LOCAL_RELAY and other local names will be sent to MAIL_HUB. Note: there is a (long standing) bug which keeps this combination from working for addresses of the form user+detail.
Names in class {L} will be delivered locally, so you MUST have aliases or .forward files for them.

For example, if you are on machine mastodon.CS.Berkeley.EDU and you have FEATURE(‘stickyhost’), the following combinations of settings will have the indicated effects:

```
email sent to....       eric                      eric@mastodon.CS.Berkeley.EDU

LOCAL_RELAY set to      mail.CS.Berkeley.EDU      (delivered locally)
mail.CS.Berkeley.EDU      (no local aliasing)       (aliasing done)

MAIL_HUB set to         mammoth.CS.Berkeley.EDU   mammoth.CS.Berkeley.EDU
mammoth.CS.Berkeley.EDU   (aliasing done)           (aliasing done)

Both LOCAL_RELAY and    mail.CS.Berkeley.EDU      mammoth.CS.Berkeley.EDU
MAIL_HUB set as above     (no local aliasing)       (aliasing done)
```

If you do not have FEATURE(‘stickyhost’) set, then LOCAL_RELAY and MAIL_HUB act identically, with MAIL_HUB taking precedence.

If you want all outgoing mail to go to a central relay site, define SMART_HOST as well. Briefly:

```
        LOCAL_RELAY applies to unqualified names (e.g., "eric").
        MAIL_HUB applies to names qualified with the name of the
                local host (e.g., "eric@mastodon.CS.Berkeley.EDU").
        SMART_HOST applies to names qualified with other hosts or
                bracketed addresses (e.g., "eric@mastodon.CS.Berkeley.EDU"
                or "eric@[127.0.0.1]").
```

However, beware that other relays (e.g., UUCP_RELAY, BITNET_RELAY, DECNET_RELAY, and FAX_RELAY) take precedence over SMART_HOST, so if you really want absolutely everything to go to a single central site you will need to unset all the other relays -- or better yet, find or build a minimal config file that does this.

For duplicate suppression to work properly, the host name is best specified with a terminal dot:

```
        define(‘MAIL_HUB’, ‘host.domain.’)
              note the trailing dot ---^
```

9.9 A better example

```
VERSIONID(‘A genuine configuration’)
OSTYPE(openbsd)
FEATURE(nouucp, ‘reject’)
FEATURE(virtusertable)
FEATURE(‘masquerade_envelope’)
MAILER(local)
MAILER(smtp)
MASQUERADE_AS(‘ifost.org.au’)
```

Notes. . .

We aren’t using UUCP, so we can reject any UUCP addresses immediately.

We should also go and create `/etc/mail/virtusertable`. It’s just an ordinary hash map made the same way as we did on page 50.

Between the “masquerade_envelope” feature and the “masquerade_as”, the name of our computer appears almost nowhere.

We have the two mailers that are almost always needed, defined very simply as above.

9.10 Exercise

Real-life .mc files

Notes. . .

1. Page 112 is fairly complete. Modify the MASQUERADE_AS option to suit your environment and create a `/etc/mail/virtusertable`. Build the configuration and try it out!
2. Map *everyone*`@co1.com` → `@co2.com`, except for `jcitizen@co1.com`, whose email should be redirected to `john.citizen@co3.com`.
3. Look at your original `sendmail.cf` file. What `.mc` file was it generated from? What features do it use? Choose a feature to add or remove (e.g. `nouucp`). Compare the new `sendmail.cf` with the old one. What changed?
4. Add `FEATURE(redirect)`. Add an alias entry for `jdoe: johnd@newplace.com REDIRECT`.

9.11 Tweaking Rulesets

- Rulesets 0 - 5 call “local” rulesets
- “Local” rulesets can be modified
- Use the name of the main ruleset

Notes. . .

For more complex configurations, you can define special rules.
The macro LOCAL_RULE_3 introduces rules that are used in canonicalizing
the names. Any modifications made here are reflected in the header.

...

This could also be used to look up hosts in a database map:

```
        LOCAL_RULE_3
        R$* < @ $+ > $*         $: $1 < @ $(hostmap $2 $) > $3
```

This map would be defined in the LOCAL_CONFIG portion, as shown below.

Similarly, LOCAL_RULE_0 can be used to introduce new parsing rules.
For example, new rules are needed to parse hostnames that you accept
via MX records. For example, you might have:

```
        LOCAL_RULE_0
        R$+ <@ host.dom.ain.>   $#uucp $@ cnmat $: $1 < @ host.dom.ain.>
```

You would use this if you had installed an MX record for cnmat.Berkeley.EDU
pointing at this host; this rule catches the message and forwards it on
using UUCP.

You can also tweak rulesets 1 and 2 using LOCAL_RULE_1 and LOCAL_RULE_2.
These rulesets are normally empty.

9.12 LOCAL_CONFIG

- Introducing other classes or maps . . .

Notes. . .

A similar macro is LOCAL_CONFIG. This introduces lines added after the boilerplate option setting but before rulesets. Do not declare rulesets in the LOCAL_CONFIG section. It can be used to declare local database maps or whatever. For example:

```
        LOCAL_CONFIG
        Khostmap hash /etc/mail/hostmap
        Kyplocal nis -m hosts.byname
```

9.13 Exercise

A sense of déja vû...

Notes...

1. On pages 44, 51 and 68 you did a number of exercises involving modifying `sendmail.cf` directly – usually to make a rule to modify something. Pick one or two of them and reformulate them to put them into a `.mc` file.

9.14 Configuration Options

confPRIVACY_FLAGS Allow EXPN, VRFY?

confSMTP_LOGIN_MSG Option `SmtpGreetingMessage`

confMIN_FREE_BLOCKS Full filesystem – stop receiving mail!

confMAX_MESSAGE_SIZE Defaults to infinite

confMATCH_GECOS From `/etc/passwd`

Notes. . .

```
There are a large number of configuration options that don't normally
need to be changed.  However, if you feel you need to tweak them, you
can define the following M4 variables.  This list is shown in four
columns:  the name you define, the default value for that definition,
the option or macro that is affected (either Ox for an option or Dx
for a macro), and a brief description.  Greater detail of the semantics
can be found in the Installation and Operations Guide.

Some options are likely to be deprecated in future versions -- that is,
the option is only included to provide back-compatibility.  These are
marked with "*".

Remember that these options are M4 variables, and hence may need to
be quoted.  In particular, arguments with commas will usually have to
be ``double quoted, like this phrase'' to avoid having the comma
confuse things.  This is common for alias file definitions and for
the read timeout.

M4 Variable Name        Configuration   Description & [Default]
================        =============   =======================
confMAILER_NAME         $n macro        [MAILER-DAEMON] The sender name used
                                        for internally generated outgoing
                                        messages.
confDOMAIN_NAME         $j macro        If defined, sets $j.  This should
                                        only be done if your system cannot
                                        determine your local domain name,
                                        and then it should be set to
                                        $w.Foo.COM, where Foo.COM is your
                                        domain name.
confCF_VERSION          $Z macro        If defined, this is appended to the
                                        configuration version name.
confLDAP_CLUSTER        ${sendmailMTACluster} macro
                                        If defined, this is the LDAP
                                        cluster to use for LDAP searches
                                        as described above in ``USING LDAP
                                        FOR ALIASES, MAPS, AND CLASSES''.
confFROM_HEADER         From:           [$?x$x <$g>$|$g$.] The format of an
                                        internally generated From: address.
confRECEIVED_HEADER     Received:
                [$?sfrom $s $.$?_($?s$|from $.$_)
                        $.$?{auth_type}(authenticated)
                        $.by $j ($v/$Z)$?r with $r$. id $i$?u
                        for $u; $|;
                        $.$b]
                                        The format of the Received: header
                                        in messages passed through this host.
                                        It is unwise to try to change this.
confCW_FILE             Fw class        [/etc/mail/local-host-names] Name
                                        of file used to get the local
                                        additions to class {w} (local host
                                        names).
confCT_FILE             Ft class        [/etc/mail/trusted-users] Name of
                                        file used to get the local additions
                                        to class {t} (trusted users).
confCR_FILE             FR class        [/etc/mail/relay-domains] Name of
                                        file used to get the local additions
                                        to class {R} (hosts allowed to relay).
confTRUSTED_USERS       Ct class        [no default] Names of users to add to
                                        the list of trusted users.  This list
                                        always includes root, uucp, and daemon.
                                        See also FEATURE(`use_ct_file').
confTRUSTED_USER        TrustedUser     [no default] Trusted user for file
                                        ownership and starting the daemon.
                                        Not to be confused with
                                        confTRUSTED_USERS (see above).
confSMTP_MAILER         -               [esmtp] The mailer name used when
                                        SMTP connectivity is required.
                                        One of "smtp", "smtp8",
                                        "esmtp", or "dsmtp".
confUUCP_MAILER         -               [uucp-old] The mailer to be used by
                                        default for bang-format recipient
                                        addresses.  See also discussion of
                                        class {U}, class {Y}, and class {Z}
                                        in the MAILER(`uucp') section.
confLOCAL_MAILER        -               [local] The mailer name used when
                                        local connectivity is required.
                                        Almost always "local".
confRELAY_MAILER        -               [relay] The default mailer name used
                                        for relaying any mail (e.g., to a
                                        BITNET_RELAY, a SMART_HOST, or
                                        whatever).  This can reasonably be
                                        "uucp-new" if you are on a
                                        UUCP-connected site.
confSEVEN_BIT_INPUT     SevenBitInput   [False] Force input to seven bits?
```

```
confEIGHT_BIT_HANDLING  EightBitMode    [pass8] 8-bit data handling
confALIAS_WAIT          AliasWait       [10m] Time to wait for alias file
                                        rebuild until you get bored and
                                        decide that the apparently pending
                                        rebuild failed.
confMIN_FREE_BLOCKS     MinFreeBlocks   [100] Minimum number of free blocks on
                                        queue filesystem to accept SMTP mail.
                                        (Prior to 8.7 this was minfree/maxsize,
                                        where minfree was the number of free
                                        blocks and maxsize was the maximum
                                        message size.  Use confMAX_MESSAGE_SIZE
                                        for the second value now.)
confMAX_MESSAGE_SIZE    MaxMessageSize  [infinite] The maximum size of messages
                                        that will be accepted (in bytes).
confBLANK_SUB           BlankSub        [.] Blank (space) substitution
                                        character.
confCON_EXPENSIVE       HoldExpensive   [False] Avoid connecting immediately
                                        to mailers marked expensive.
confCHECKPOINT_INTERVAL CheckpointInterval
                                        [10] Checkpoint queue files every N
                                        recipients.
confDELIVERY_MODE       DeliveryMode    [background] Default delivery mode.
confERROR_MODE          ErrorMode       [print] Error message mode.
confERROR_MESSAGE       ErrorHeader     [undefined] Error message header/file.
confSAVE_FROM_LINES     SaveFromLine    Save extra leading From_ lines.
confTEMP_FILE_MODE      TempFileMode    [0600] Temporary file mode.
confMATCH_GECOS         MatchGECOS      [False] Match GECOS field.
confMAX_HOP             MaxHopCount     [25] Maximum hop count.
confIGNORE_DOTS*        IgnoreDots      [False; always False in -bs or -bd
                                        mode] Ignore dot as terminator for
                                        incoming messages?
confBIND_OPTS           ResolverOptions [undefined] Default options for DNS
                                        resolver.
confMIME_FORMAT_ERRORS* SendMimeErrors  [True] Send error messages as MIME-
                                        encapsulated messages per RFC 1344.
confFORWARD_PATH        ForwardPath     [$z/.forward.$w:$z/.forward]
                                        The colon-separated list of places to
                                        search for .forward files.  N.B.: see
                                        the Security Notes section.
confMCI_CACHE_SIZE      ConnectionCacheSize
                                        [2] Size of open connection cache.
confMCI_CACHE_TIMEOUT   ConnectionCacheTimeout
                                        [5m] Open connection cache timeout.
confHOST_STATUS_DIRECTORY HostStatusDirectory
                                        [undefined] If set, host status is kept
                                        on disk between sendmail runs in the
                                        named directory tree.  This need not be
                                        a full pathname, in which case it is
                                        interpreted relative to the queue
                                        directory.
```

confSINGLE_THREAD_DELIVERY	SingleThreadDelivery	[False] If this option and the HostStatusDirectory option are both set, single thread deliveries to other hosts. That is, don't allow any two sendmails on this host to connect simultaneously to any other single host. This can slow down delivery in some cases, in particular since a cached but otherwise idle connection to a host will prevent other sendmails from connecting to the other host.
confUSE_ERRORS_TO*	UseErrorsTo	[False] Use the Errors-To: header to deliver error messages. This should not be necessary because of general acceptance of the envelope/header distinction.
confLOG_LEVEL	LogLevel	[9] Log level.
confME_TOO	MeToo	[True] Include sender in group expansions. This option is deprecated and will be removed from a future version.
confCHECK_ALIASES	CheckAliases	[False] Check RHS of aliases when running newaliases. Since this does DNS lookups on every address, it can slow down the alias rebuild process considerably on large alias files.
confOLD_STYLE_HEADERS*	OldStyleHeaders	[True] Assume that headers without special chars are old style.
confPRIVACY_FLAGS	PrivacyOptions	[authwarnings] Privacy flags.
confCOPY_ERRORS_TO	PostmasterCopy	[undefined] Address for additional copies of all error messages.
confQUEUE_FACTOR	QueueFactor	[600000] Slope of queue-only function.
confQUEUE_FILE_MODE	QueueFileMode	[undefined] Default permissions for queue files (octal). If not set, sendmail uses 0600 unless its real and effective uid are different in which case it uses 0644.
confDONT_PRUNE_ROUTES	DontPruneRoutes	[False] Don't prune down route-addr syntax addresses to the minimum possible.
confSAFE_QUEUE*	SuperSafe	[True] Commit all messages to disk before forking.
confTO_INITIAL	Timeout.initial	[5m] The timeout waiting for a response on the initial connect.
confTO_CONNECT	Timeout.connect	[0] The timeout waiting for an initial connect() to complete. This can only shorten connection timeouts; the kernel silently enforces an absolute maximum (which varies depending on the system).

```
confTO_ICONNECT         Timeout.iconnect
                                        [undefined] Like Timeout.connect, but
                                        applies only to the very first attempt
                                        to connect to a host in a message.
                                        This allows a single very fast pass
                                        followed by more careful delivery
                                        attempts in the future.
confTO_ACONNECT         Timeout.aconnect
                                        [0] The overall timeout waiting for
                                        all connection for a single delivery
                                        attempt to succeed.  If 0, no overall
                                        limit is applied.
confTO_HELO             Timeout.helo    [5m] The timeout waiting for a response
                                        to a HELO or EHLO command.
confTO_MAIL             Timeout.mail    [10m] The timeout waiting for a
                                        response to the MAIL command.
confTO_RCPT             Timeout.rcpt    [1h] The timeout waiting for a response
                                        to the RCPT command.
confTO_DATAINIT         Timeout.datainit
                                        [5m] The timeout waiting for a 354
                                        response from the DATA command.
confTO_DATABLOCK        Timeout.datablock
                                        [1h] The timeout waiting for a block
                                        during DATA phase.
confTO_DATAFINAL        Timeout.datafinal
                                        [1h] The timeout waiting for a response
                                        to the final "." that terminates a
                                        message.
confTO_RSET             Timeout.rset    [5m] The timeout waiting for a response
                                        to the RSET command.
confTO_QUIT             Timeout.quit    [2m] The timeout waiting for a response
                                        to the QUIT command.
confTO_MISC             Timeout.misc    [2m] The timeout waiting for a response
                                        to other SMTP commands.
confTO_COMMAND          Timeout.command [1h] In server SMTP, the timeout
                                        waiting for a command to be issued.
confTO_IDENT            Timeout.ident   [5s] The timeout waiting for a
                                        response to an IDENT query.
confTO_FILEOPEN         Timeout.fileopen
                                        [60s] The timeout waiting for a file
                                        (e.g., :include: file) to be opened.
confTO_LHLO             Timeout.lhlo    [2m] The timeout waiting for a response
                                        to an LMTP LHLO command.
confTO_AUTH             Timeout.auth    [10m] The timeout waiting for a
                                        response in an AUTH dialogue.
confTO_STARTTLS         Timeout.starttls
                                        [1h] The timeout waiting for a
                                        response to an SMTP STARTTLS command.
confTO_CONTROL          Timeout.control
                                        [2m] The timeout for a complete
                                        control socket transaction to complete.
confTO_QUEUERETURN      Timeout.queuereturn
                                        [5d] The timeout before a message is
                                        returned as undeliverable.
confTO_QUEUERETURN_NORMAL
                        Timeout.queuereturn.normal
                                        [undefined] As above, for normal
                                        priority messages.
confTO_QUEUERETURN_URGENT
                        Timeout.queuereturn.urgent
                                        [undefined] As above, for urgent
                                        priority messages.
confTO_QUEUERETURN_NONURGENT
                        Timeout.queuereturn.non-urgent
                                        [undefined] As above, for non-urgent
                                        (low) priority messages.
confTO_QUEUEWARN        Timeout.queuewarn
                                        [4h] The timeout before a warning
                                        message is sent to the sender telling
                                        them that the message has been
                                        deferred.
confTO_QUEUEWARN_NORMAL Timeout.queuewarn.normal
                                        [undefined] As above, for normal
                                        priority messages.
confTO_QUEUEWARN_URGENT Timeout.queuewarn.urgent
                                        [undefined] As above, for urgent
                                        priority messages.
confTO_QUEUEWARN_NONURGENT
                        Timeout.queuewarn.non-urgent
                                        [undefined] As above, for non-urgent
                                        (low) priority messages.
confTO_HOSTSTATUS       Timeout.hoststatus
                                        [30m] How long information about host
                                        statuses will be maintained before it
                                        is considered stale and the host should
                                        be retried.  This applies both within
                                        a single queue run and to persistent
                                        information (see below).
confTO_RESOLVER_RETRANS Timeout.resolver.retrans
                                        [varies] Sets the resolver's
                                        retransmition time interval (in
                                        seconds).  Sets both
                                        Timeout.resolver.retrans.first and
                                        Timeout.resolver.retrans.normal.
confTO_RESOLVER_RETRANS_FIRST  Timeout.resolver.retrans.first
                                        [varies] Sets the resolver's
                                        retransmition time interval (in
                                        seconds) for the first attempt to
                                        deliver a message.
confTO_RESOLVER_RETRANS_NORMAL  Timeout.resolver.retrans.normal
```

```
                                 [varies] Sets the resolver's
                                 retransmition time interval (in
                                 seconds) for all resolver lookups
                                 except the first delivery attempt.
confTO_RESOLVER_RETRY   Timeout.resolver.retry
                                 [varies] Sets the number of times
                                 to retransmit a resolver query.
                                 Sets both
                                 Timeout.resolver.retry.first and
                                 Timeout.resolver.retry.normal.
confTO_RESOLVER_RETRY_FIRST  Timeout.resolver.retry.first
                                 [varies] Sets the number of times
                                 to retransmit a resolver query for
                                 the first attempt to deliver a
                                 message.
confTO_RESOLVER_RETRY_NORMAL  Timeout.resolver.retry.normal
                                 [varies] Sets the number of times
                                 to retransmit a resolver query for
                                 all resolver lookups except the
                                 first delivery attempt.
confTIME_ZONE           TimeZoneSpec     [USE_SYSTEM] Time zone info -- can be
                                 USE_SYSTEM to use the system's idea,
                                 USE_TZ to use the user's TZ envariable,
                                 or something else to force that value.
confDEF_USER_ID         DefaultUser      [1:1] Default user id.
confUSERDB_SPEC         UserDatabaseSpec
                                 [undefined] User database
                                 specification.
confFALLBACK_MX         FallbackMXhost   [undefined] Fallback MX host.
confTRY_NULL_MX_LIST    TryNullMXList    [False] If this host is the best MX
                                 for a host and other arrangements
                                 haven't been made, try connecting
                                 to the host directly; normally this
                                 would be a config error.
confQUEUE_LA            QueueLA          [varies] Load average at which
                                 queue-only function kicks in.
                                 Default values is (8 * numproc)
                                 where numproc is the number of
                                 processors online (if that can be
                                 determined).
confREFUSE_LA           RefuseLA         [varies] Load average at which
                                 incoming SMTP connections are
                                 refused.  Default values is (12 *
                                 numproc) where numproc is the
                                 number of processors online (if
                                 that can be determined).
confDELAY_LA            DelayLA          [0] Load average at which sendmail
                                 will sleep for one second on most
                                 SMTP commands and before accepting
                                 connections.  0 means no limit.
```

```
confMAX_ALIAS_RECURSION MaxAliasRecursion
                                        [10] Maximum depth of alias recursion.
confMAX_DAEMON_CHILDREN MaxDaemonChildren
                                        [undefined] The maximum number of
                                        children the daemon will permit.  After
                                        this number, connections will be
                                        rejected.  If not set or <= 0, there is
                                        no limit.
confMAX_HEADERS_LENGTH  MaxHeadersLength
                                        [32768] Maximum length of the sum
                                        of all headers.
confMAX_MIME_HEADER_LENGTH  MaxMimeHeaderLength
                                        [undefined] Maximum length of
                                        certain MIME header field values.
confCONNECTION_RATE_THROTTLE ConnectionRateThrottle
                                        [undefined] The maximum number of
                                        connections permitted per second per
                                        daemon.  After this many connections
                                        are accepted, further connections
                                        will be delayed.  If not set or <= 0,
                                        there is no limit.
confWORK_RECIPIENT_FACTOR
                        RecipientFactor [30000] Cost of each recipient.
confSEPARATE_PROC       ForkEachJob     [False] Run all deliveries in a
                                        separate process.
confWORK_CLASS_FACTOR   ClassFactor     [1800] Priority multiplier for class.
confWORK_TIME_FACTOR    RetryFactor     [90000] Cost of each delivery attempt.
confQUEUE_SORT_ORDER    QueueSortOrder  [Priority] Queue sort algorithm:
                                        Priority, Host, Filename, Random,
                                        Modification, or Time.
confMIN_QUEUE_AGE       MinQueueAge     [0] The minimum amount of time a job
                                        must sit in the queue between queue
                                        runs.  This allows you to set the
                                        queue run interval low for better
                                        responsiveness without trying all
                                        jobs in each run.
confDEF_CHAR_SET        DefaultCharSet  [unknown-8bit] When converting
                                        unlabeled 8 bit input to MIME, the
                                        character set to use by default.
confSERVICE_SWITCH_FILE ServiceSwitchFile
                                        [/etc/mail/service.switch] The file
                                        to use for the service switch on
                                        systems that do not have a
                                        system-defined switch.
confHOSTS_FILE          HostsFile       [/etc/hosts] The file to use when doing
                                        "file" type access of hosts names.
confDIAL_DELAY          DialDelay       [0s] If a connection fails, wait this
                                        long and try again.  Zero means "don't
                                        retry".  This is to allow "dial on
                                        demand" connections to have enough time
                                        to complete a connection.
confNO_RCPT_ACTION      NoRecipientAction
                                        [none] What to do if there are no legal
                                        recipient fields (To:, Cc: or Bcc:)
                                        in the message.  Legal values can
                                        be "none" to just leave the
                                        nonconforming message as is, "add-to"
                                        to add a To: header with all the
                                        known recipients (which may expose
                                        blind recipients), "add-apparently-to"
                                        to do the same but use Apparently-To:
                                        instead of To: (strongly discouraged
                                        in accordance with IETF standards),
                                        "add-bcc" to add an empty Bcc:
                                        header, or "add-to-undisclosed" to
                                        add the header
                                        ``To: undisclosed-recipients:;''.
confSAFE_FILE_ENV       SafeFileEnvironment
                                        [undefined] If set, sendmail will do a
                                        chroot() into this directory before
                                        writing files.
confCOLON_OK_IN_ADDR    ColonOkInAddr   [True unless Configuration Level > 6]
                                        If set, colons are treated as a regular
                                        character in addresses.  If not set,
                                        they are treated as the introducer to
                                        the RFC 822 "group" syntax.  Colons are
                                        handled properly in route-addrs.  This
                                        option defaults on for V5 and lower
                                        configuration files.
confMAX_QUEUE_RUN_SIZE  MaxQueueRunSize [0] If set, limit the maximum size of
                                        any given queue run to this number of
                                        entries.  Essentially, this will stop
                                        reading each queue directory after this
                                        number of entries are reached; it does
                                        _not_ pick the highest priority jobs,
                                        so this should be as large as your
                                        system can tolerate.  If not set, there
                                        is no limit.
confMAX_QUEUE_CHILDREN  MaxQueueChildren
                                        [undefined] Limits the maximum number
                                        of concurrent queue runners active.
                                        This is to keep system resources used
                                        within a reasonable limit.  Relates to
                                        Queue Groups and ForkAllJobs.
confMAX_RUNNERS_PER_QUEUE       MaxRunnersPerQueue
                                        [1] Only active when MaxQueueChildren
                                        defined.  Controls the maximum number
                                        of queue runners (aka queue children)
                                        active at the same time in a work
                                        group.  See also MaxQueueChildren.
```

```
confDONT_EXPAND_CNAMES  DontExpandCnames
                                        [False] If set, $[ ... $] lookups that
                                        do DNS based lookups do not expand
                                        CNAME records.  This currently violates
                                        the published standards, but the IETF
                                        seems to be moving toward legalizing
                                        this.  For example, if "FTP.Foo.ORG"
                                        is a CNAME for "Cruft.Foo.ORG", then
                                        with this option set a lookup of
                                        "FTP" will return "FTP.Foo.ORG"; if
                                        clear it returns "Cruft.FOO.ORG".  N.B.
                                        you may not see any effect until your
                                        downstream neighbors stop doing CNAME
                                        lookups as well.
confFROM_LINE           UnixFromLine    [From $g $d] The From_ line used
                                        when sending to files or programs.
confSINGLE_LINE_FROM_HEADER  SingleLineFromHeader
                                        [False] From: lines that have
                                        embedded newlines are unwrapped
                                        onto one line.
confALLOW_BOGUS_HELO    AllowBogusHELO  [False] Allow HELO SMTP command that
                                        does not include a host name.
confMUST_QUOTE_CHARS    MustQuoteChars  [.'] Characters to be quoted in a full
                                        name phrase (@,;:\()[] are automatic).
confOPERATORS           OperatorChars   [.:%@!^/[]+] Address operator
                                        characters.
confSMTP_LOGIN_MSG      SmtpGreetingMessage
                                        [$j Sendmail $v/$Z; $b]
                                        The initial (spontaneous) SMTP
                                        greeting message.  The word "ESMTP"
                                        will be inserted between the first and
                                        second words to convince other
                                        sendmails to try to speak ESMTP.
confDONT_INIT_GROUPS    DontInitGroups  [False] If set, the initgroups(3)
                                        routine will never be invoked.  You
                                        might want to do this if you are
                                        running NIS and you have a large group
                                        map, since this call does a sequential
                                        scan of the map; in a large site this
                                        can cause your ypserv to run
                                        essentially full time.  If you set
                                        this, agents run on behalf of users
                                        will only have their primary
                                        (/etc/passwd) group permissions.
confUNSAFE_GROUP_WRITES UnsafeGroupWrites
                                        [False] If set, group-writable
                                        :include: and .forward files are
                                        considered "unsafe", that is, programs
                                        and files cannot be directly referenced
                                        from such files.  World-writable files
                                        are always considered unsafe.
confCONNECT_ONLY_TO     ConnectOnlyTo   [undefined] override connection
                                        address (for testing).
confCONTROL_SOCKET_NAME ControlSocketName
                                        [undefined] Control socket for daemon
                                        management.
confDOUBLE_BOUNCE_ADDRESS  DoubleBounceAddress
                                        [postmaster] If an error occurs when
                                        sending an error message, send that
                                        "double bounce" error message to this
                                        address.  If it expands to an empty
                                        string, double bounces are dropped.
confDEAD_LETTER_DROP    DeadLetterDrop  [undefined] Filename to save bounce
                                        messages which could not be returned
                                        to the user or sent to postmaster.
                                        If not set, the queue file will
                                        be renamed.
confRRT_IMPLIES_DSN     RrtImpliesDsn   [False] Return-Receipt-To: header
                                        implies DSN request.
confRUN_AS_USER         RunAsUser       [undefined] If set, become this user
                                        when reading and delivering mail.
                                        Causes all file reads (e.g., .forward
                                        and :include: files) to be done as
                                        this user.  Also, all programs will
                                        be run as this user, and all output
                                        files will be written as this user.
                                        Intended for use only on firewalls
                                        where users do not have accounts.
confMAX_RCPTS_PER_MESSAGE  MaxRecipientsPerMessage
                                        [infinite] If set, allow no more than
                                        the specified number of recipients in
                                        an SMTP envelope.  Further recipients
                                        receive a 452 error code (i.e., they
                                        are deferred for the next delivery
                                        attempt).
confBAD_RCPT_THROTTLE   BadRcptThrottle [infinite] If set and more than the
                                        specified number of recipients in an
                                        envelope are rejected, sleep for one
                                        second after each rejected RCPT
                                        command.
confDONT_PROBE_INTERFACES  DontProbeInterfaces
                                        [False] If set, sendmail will _not_
                                        insert the names and addresses of any
                                        local interfaces into class {w}
                                        (list of known "equivalent" addresses).
                                        If you set this, you must also include
                                        some support for these addresses (e.g.,
                                        in a mailertable entry) -- otherwise,
                                        mail to addresses in this list will
                                        bounce with a configuration error.
                                        If set to "loopback" (without
                                        quotes), sendmail will skip
                                        loopback interfaces (e.g., "lo0").
confPID_FILE            PidFile         [system dependent] Location of pid
                                        file.
confPROCESS_TITLE_PREFIX  ProcessTitlePrefix
                                        [undefined] Prefix string for the
                                        process title shown on 'ps' listings.
confDONT_BLAME_SENDMAIL DontBlameSendmail
                                        [safe] Override sendmail's file
                                        safety checks.  This will definitely
                                        compromise system security and should
                                        not be used unless absolutely
                                        necessary.
confREJECT_MSG          -               [550 Access denied] The message
                                        given if the access database contains
                                        REJECT in the value portion.
confRELAY_MSG           -               [550 Relaying denied] The message
                                        given if an unauthorized relaying
                                        attempt is rejected.
confDF_BUFFER_SIZE      DataFileBufferSize
                                        [4096] The maximum size of a
                                        memory-buffered data (df) file
                                        before a disk-based file is used.
confXF_BUFFER_SIZE      XScriptFileBufferSize
                                        [4096] The maximum size of a
                                        memory-buffered transcript (xf)
                                        file before a disk-based file is
                                        used.
confAUTH_MECHANISMS     AuthMechanisms  [GSSAPI KERBEROS_V4 DIGEST-MD5
                                        CRAM-MD5] List of authentication
                                        mechanisms for AUTH (separated by
                                        spaces).  The advertised list of
                                        authentication mechanisms will be the
                                        intersection of this list and the list
                                        of available mechanisms as determined
                                        by the CYRUS SASL library.
confDEF_AUTH_INFO       DefaultAuthInfo [undefined] Name of file that contains
                                        authentication information for
                                        outgoing connections.  This file must
                                        contain the user id, the authorization
                                        id, the password (plain text), the
                                        realm to use, and the list of
                                        mechanisms to try, each on a separate
                                        line and must be readable by root (or
                                        the trusted user) only.  If no realm
                                        is specified, $j is used.  If no
                                        mechanisms are given in the file,
                                        AuthMechanisms is used.  Notice: this
                                        option is deprecated and will be
```

		removed in future versions; it doesn't work for the MSP since it can't read the file. Use the authinfo ruleset instead. See also the section SMTP AUTHENTICATION.
confAUTH_OPTIONS	AuthOptions	[undefined] If this option is 'A' then the AUTH= parameter for the MAIL FROM command is only issued when authentication succeeded. Other values (which should be listed one after the other without any intervening characters except for space or comma) are a, c, d, f, p, and y. See doc/op/op.me for details.
confAUTH_MAX_BITS	AuthMaxBits	[INT_MAX] Limit the maximum encryption strength for the security layer in SMTP AUTH (SASL). Default is essentially unlimited.
confTLS_SRV_OPTIONS	TLSSrvOptions	If this option is 'V' no client verification is performed, i.e., the server doesn't ask for a certificate.
confLDAP_DEFAULT_SPEC	LDAPDefaultSpec	[undefined] Default map specification for LDAP maps. The value should only contain LDAP specific settings such as "-h host -p port -d bindDN", etc. The settings will be used for all LDAP maps unless they are specified in the individual map specification ('K' command).
confCACERT_PATH	CACERTPath	[undefined] Path to directory with certs of CAs.
confCACERT	CACERTFile	[undefined] File containing one CA cert.
confSERVER_CERT	ServerCertFile	[undefined] File containing the cert of the server, i.e., this cert is used when sendmail acts as server.
confSERVER_KEY	ServerKeyFile	[undefined] File containing the private key belonging to the server cert.
confCLIENT_CERT	ClientCertFile	[undefined] File containing the cert of the client, i.e., this cert is used when sendmail acts as client.
confCLIENT_KEY	ClientKeyFile	[undefined] File containing the private key belonging to the client cert.

confDH_PARAMETERS	DHParameters	[undefined] File containing the DH parameters.
confRAND_FILE	RandFile	[undefined] File containing random data (use prefix file:) or the name of the UNIX socket if EGD is used (use prefix egd:). STARTTLS requires this option if the compile flag HASURANDOM is not set (see sendmail/README).
confNICE_QUEUE_RUN	NiceQueueRun	[undefined] If set, the priority of queue runners is set the given value (nice(3)).
confDIRECT_SUBMISSION_MODIFIERS	DirectSubmissionModifiers	[undefined] Defines {daemon_flags} for direct submissions.
confUSE_MSP	UseMSP	[false] Use as mail submission program, see sendmail/SECURITY.
confDELIVER_BY_MIN	DeliverByMin	[0] Minimum time for Deliver By SMTP Service Extension (RFC 2852).
confSHARED_MEMORY_KEY	SharedMemoryKey	[0] Key for shared memory.
confFAST_SPLIT	FastSplit	[1] If set to a value greater than zero, the initial MX lookups on addresses is suppressed when they are sorted which may result in faster envelope splitting. If the mail is submitted directly from the command line, then the value also limits the number of processes to deliver the envelopes.
confMAILBOX_DATABASE	MailboxDatabase	[pw] Type of lookup to find information about local mailboxes.
confDEQUOTE_OPTS	-	[empty] Additional options for the dequote map.
confINPUT_MAIL_FILTERS	InputMailFilters	A comma separated list of filters which determines which filters and the invocation sequence are contacted for incoming SMTP messages. If none are set, no filters will be contacted.
confMILTER_LOG_LEVEL	Milter.LogLevel	[9] Log level for input mail filter actions, defaults to LogLevel.
confMILTER_MACROS_CONNECT	Milter.macros.connect	[empty] Macros to transmit to milters when a session connection starts.
confMILTER_MACROS_HELO	Milter.macros.helo	[empty] Macros to transmit to milters after HELO command.
confMILTER_MACROS_ENVFROM	Milter.macros.envfrom	[empty] Macros to transmit to milters

```
                                        after MAIL FROM command.
confMILTER_MACROS_ENVRCPT       Milter.macros.envrcpt
                                        [empty] Macros to transmit to milters
                                        after RCPT TO command.
```

See also the description of OSTYPE for some parameters that can be tweaked (generally pathnames to mailers).

ClientPortOptions and DaemonPortOptions are special cases since multiple clients/daemons can be defined. This can be done via

```
        CLIENT_OPTIONS(`field1=value1,field2=value2,...')
        DAEMON_OPTIONS(`field1=value1,field2=value2,...')
```

Note that multiple CLIENT_OPTIONS() commands (and therefore multiple ClientPortOptions settings) are allowed in order to give settings for each protocol family (e.g., one for Family=inet and one for Family=inet6). A restriction placed on one family only affects outgoing connections on that particular family.

If DAEMON_OPTIONS is not used, then the default is

```
        DAEMON_OPTIONS(`Port=smtp, Name=MTA')
        DAEMON_OPTIONS(`Port=587, Name=MSA, M=E')
```

If you use one DAEMON_OPTIONS macro, it will alter the parameters of the first of these. The second will still be defaulted; it represents a "Message Submission Agent" (MSA) as defined by RFC 2476 (see below). To turn off the default definition for the MSA, use FEATURE(`no_default_msa') (see also FEATURES). If you use additional DAEMON_OPTIONS macros, they will add additional daemons.

Example 1: To change the port for the SMTP listener, while still using the MSA default, use

```
        DAEMON_OPTIONS(`Port=925, Name=MTA')
```

Example 2: To change the port for the MSA daemon, while still using the default SMTP port, use

```
        FEATURE(`no_default_msa')
        DAEMON_OPTIONS(`Name=MTA')
        DAEMON_OPTIONS(`Port=987, Name=MSA, M=E')
```

Note that if the first of those DAEMON_OPTIONS lines were omitted, then there would be no listener on the standard SMTP port.

Example 3: To listen on both IPv4 and IPv6 interfaces, use

```
        DAEMON_OPTIONS(`Name=MTA-v4, Family=inet')
        DAEMON_OPTIONS(`Name=MTA-v6, Family=inet6')
```

A "Message Submission Agent" still uses all of the same rulesets for processing the message (and therefore still allows message rejection via the check_* rulesets). In accordance with the RFC, the MSA will ensure that all domains in the envelope are fully qualified if the message is relayed to another MTA. It will also enforce the normal address syntax rules and log error messages. Additionally, by using the M=a modifier you can require authentication before messages are accepted by the MSA. Notice: Do NOT use the 'a' modifier on a public accessible MTA!
Finally, the M=E modifier shown above disables ETRN as required by RFC 2476.

Mail filters can be defined using the INPUT_MAIL_FILTER() and MAIL_FILTER() commands:

```
        INPUT_MAIL_FILTER(`sample', `S=local:/var/run/f1.sock')
        MAIL_FILTER(`myfilter', `S=inet:3333@localhost')
```

The INPUT_MAIL_FILTER() command causes the filter(s) to be called in the same order they were specified by also setting confINPUT_MAIL_FILTERS. A filter can be defined without adding it to the input filter list by using MAIL_FILTER() instead of INPUT_MAIL_FILTER() in your .mc file. Alternatively, you can reset the list of filters and their order by setting confINPUT_MAIL_FILTERS option after all INPUT_MAIL_FILTER() commands in your .mc file.

9.15 MAILERS

You probably want:

- MAILER(local)
- MAILER(smtp)

Notes. . .

```
There are fewer mailers supported in this version than the previous
version, owing mostly to a simpler world.  As a general rule, put the
MAILER definitions last in your .mc file.

local           The local and prog mailers.  You will almost always
                need these; the only exception is if you relay ALL
                your mail to another site.  This mailer is included
                automatically.

smtp            The Simple Mail Transport Protocol mailer.  This does
                not hide hosts behind a gateway or another other
                such hack; it assumes a world where everyone is
                running the name server.  This file actually defines
                five mailers: "smtp" for regular (old-style) SMTP to
                other servers, "esmtp" for extended SMTP to other
                servers, "smtp8" to do SMTP to other servers without
                converting 8-bit data to MIME (essentially, this is
                your statement that you know the other end is 8-bit
                clean even if it doesn't say so), "dsmtp" to do on
                demand delivery, and "relay" for transmission to the
                RELAY_HOST, LUSER_RELAY, or MAIL_HUB.

uucp            The UNIX-to-UNIX Copy Program mailer.  Actually, this
                defines two mailers, "uucp-old" (a.k.a. "uucp") and
                "uucp-new" (a.k.a. "suucp").  The latter is for when you
                know that the UUCP mailer at the other end can handle
                multiple recipients in one transfer.  If the smtp mailer
                is included in your configuration, two other mailers
                ("uucp-dom" and "uucp-uudom") are also defined [warning: you
                MUST specify MAILER(`smtp') before MAILER(`uucp')].  When you
                include the uucp mailer, sendmail looks for all names in
                class {U} and sends them to the uucp-old mailer; all
                names in class {Y} are sent to uucp-new; and all
                names in class {Z} are sent to uucp-uudom.  Note that
                this is a function of what version of rmail runs on
                the receiving end, and hence may be out of your control.
                See the section below describing UUCP mailers in more
```

detail.

usenet

Usenet (network news) delivery. If this is specified, an extra rule is added to ruleset 0 that forwards all local email for users named ‘‘group.usenet’’ to the ‘‘inews’’ program. Note that this works for all groups, and may be considered a security problem.

fax

Facsimile transmission. This is experimental and based on Sam Leffler’s HylaFAX software. For more information, see http://www.hylafax.org/.

pop

Post Office Protocol.

procmail

An interface to procmail (does not come with sendmail). This is designed to be used in mailertables. For example, a common question is "how do I forward all mail for a given domain to a single person?". If you have this mailer defined, you could set up a mailertable reading:

```
        host.com        procmail:/etc/procmailrcs/host.com
```

with the file /etc/procmailrcs/host.com reading:

```
        :0      # forward mail for host.com
        ! -oi -f $1 person@other.host
```

This would arrange for (anything)@host.com to be sent to person@other.host. Within the procmail script, $1 is the name of the sender and $2 is the name of the recipient. If you use this with FEATURE(‘local_procmail’), the FEATURE should be listed first.

Of course there are other ways to solve this particular problem, e.g., a catch-all entry in a virtusertable.

mail11

The DECnet mail11 mailer, useful only if you have the mail11 program from gatekeeper.dec.com:/pub/DEC/gwtools (and DECnet, of course). This is for Phase IV DECnet support; if you have Phase V at your site you may have additional problems.

phquery

The phquery program. This is somewhat counterintuitively referenced as the "ph" mailer internally. It can be used to do CCSO name server lookups. The phquery program, which this mailer uses, is distributed with the ph client.

cyrus

The cyrus and cyrusbb mailers. The cyrus mailer delivers to a local cyrus user. this mailer can make use of the "user+detail@local.host" syntax (see

FEATURE(‘preserve_local_plus_detail’)); it will deliver the mail to the user’s "detail" mailbox if the mailbox’s ACL permits. The cyrusbb mailer delivers to a system-wide cyrus mailbox if the mailbox’s ACL permits. The cyrus mailer must be defined after the local mailer.

qpage A mailer for QuickPage, a pager interface. See http://www.qpage.org/ for further information.

The local mailer accepts addresses of the form "user+detail", where the "+detail" is not used for mailbox matching but is available to certain local mail programs (in particular, see FEATURE(‘local_procmail’)). For example, "eric", "eric+sendmail", and "eric+sww" all indicate the same user, but additional arguments <null>, "sendmail", and "sww" may be provided for use in sorting mail.

9.16 DOMAINS

- For big sites only
- Centralises names for relay servers
- Not necessary at all

Notes...

From cf/README:

```
You will probably want to collect domain-dependent defines into one
file, referenced by the DOMAIN macro.  For example, the Berkeley
domain file includes definitions for several internal distinguished
hosts:

UUCP_RELAY      The host that will accept UUCP-addressed email.
                If not defined, all UUCP sites must be directly
                connected.
BITNET_RELAY    The host that will accept BITNET-addressed email.
                If not defined, the .BITNET pseudo-domain won’t work.
DECNET_RELAY    The host that will accept DECNET-addressed email.
                If not defined, the .DECNET pseudo-domain and addresses
                of the form node::user will not work.
FAX_RELAY       The host that will accept mail to the .FAX pseudo-domain.
                The "fax" mailer overrides this value.
LOCAL_RELAY     The site that will handle unqualified names -- that
                is, names without an @domain extension.
                Normally MAIL_HUB is preferred for this function.
                LOCAL_RELAY is mostly useful in conjunction with
                FEATURE(‘stickyhost’) -- see the discussion of
                stickyhost below.  If not set, they are assumed to
                belong on this machine.  This allows you to have a
                central site to store a company- or department-wide
                alias database.  This only works at small sites,
                and only with some user agents.
LUSER_RELAY     The site that will handle lusers -- that is, apparently
                local names that aren’t local accounts or aliases.  To
                specify a local user instead of a site, set this to
                ‘‘local:username’’.

Any of these can be either ‘‘mailer:hostname’’ (in which case the
mailer is the internal mailer name, such as ‘‘uucp-new’’ and the hostname
is the name of the host as appropriate for that mailer) or just a
‘‘hostname’’, in which case a default mailer type (usually ‘‘relay’’,
a variant on SMTP) is used.  WARNING: if you have a wildcard MX
```

record matching your domain, you probably want to define these to have a trailing dot so that you won't get the mail diverted back to yourself.

The domain file can also be used to define a domain name, if needed (using "DD<domain>") and set certain site-wide features. If all hosts at your site masquerade behind one email name, you could also use MASQUERADE_AS here.

You do not have to define a domain -- in particular, if you are a single machine sitting off somewhere, it is probably more work than it's worth. This is just a mechanism for combining "domain dependent knowledge" into one place.

Chapter 10

Reducing SPAM

10.1 Statistics for my servers

1078	**Total rejected messages**
147	HELO failures
269	Non-existant domains
469	Common invalid mailboxes
190	Other non-existant names
2	Yahoo oddity
1	Misconfiguration

Notes...

The above were for the period March 16 - Mar 22, 2004, on my server `segala.ifost.org.au`. During the same time period, it accepted 1117 valid messages – I don't have any good statistics about how many of those were worth delivering or not.

1078	**Total rejected messages**
7	The HELO banner could never be a valid hostname, since it had spaces or other illegal characters.
140	The HELO banner was just a short hostname, not a fully qualified hostname.
269	The FROM address had a domain which did not resolve to having any kind of MX record.
159	The TO address was `info@...`
194	The TO address was `admin@...`
31	The TO address was `ceo@...`
73	The TO address was `president@...`
12	The TO address was `customerservice@...`
190	The TO address was some other address that didn't exist.
2	The message came from a yahoo account, but not via a yahoo mail server.
1	Should have been delivered correctly, but wasn't because of a misconfiguration on my part.

10.2 What sendmail does automatically

- Reject unresolvable domains
- Reject unqualified names (user, but no domain)
- Reject invalid HELOs

Notes. . .

Normally, the auto-of-the-box configuration for *sendmail* does exactly what you want in these areas. And this will help minimize SPAM being delivered to your users.

But if you have a weird configuration (e.g. split DNS, and an internal domain name that doesn't exist externally), then you might need to turn off the rejection of unresolvable domains. You can do this with the following line in your `.mc` file:

```
FEATURE(‘accept_unresolvable_domains’)
```

Similarly, to stop the rejection of unqualified names:

```
FEATURE(‘accept_unqualified_senders’)
```

As far as I can tell, there is no way to tell *sendmail* to let broken mail servers connect without specifying a proper HELO greeting. Which is fair enough, really.

10.3 Blacklists

- A DNS domain
- Keeps track of IP addresses that send SPAM
- Many organisations maintain blacklists

Notes. . .

Another simple way of reducing (but not eliminating) SPAM is to block messages coming from IP addresses from which SPAM has come before. Done on a global scale – if you send SPAM, within a few hours or days you will not be able to send email again to most mail servers on the planet. At least, that's the idea.

10.4 Using a blacklist manually

- You get a connect from IP address `A.B.C.D`
- Look up the A record for `D.C.B.A.`*relays.ordb.org*
- If you get a response – it's a SPAM domain, see TXT record for the reason
- If you don't get a response, it's not a known spammer

Notes. . .

You can test this on a command line on most modern Unix systems like this:

```
host -t a 217.47.87.217.relays.ordb.org
```

or for older systems (and on Windows NT/2k/XP)

```
nslookup 217.47.87.217.relays.ordb.org
```

And if you get a match:

```
host -t txt 217.47.87.217.relays.ordb.org
```

or

```
nslookup -type=txt 217.47.87.217.relays.ordb.org
```

10.5 Very quick blacklist exercise

Slightly contrived, but helpful

Notes. . .

1. Which of these IP addresses should you reject email from? Why?

   - 65.54.166.99
   - 127.0.0.2
   - 207.106.6.148

10.6 Lists I use / have used

Subdomain	Purpose
`relays.ordb.org`	Open relay servers
`opm.blitzed.org`	Open proxies
`lists.dsbl.org`	Unsecure servers
`spl.spamhaus.org`	Known spammers
`cbl.abuseat.org`	Worms, trojans, etc.

Notes...

This information is partly lifted from `www.declude.com/Junkmail/support/ip4r.htm`

I'm a bit of a cheap-skate, so I haven't tried any of the commercial-access lists yet.

One annoying issue with `list.dsbl.org` is that it blocks dynamically-allocated address blocks. So a customer or supplier using (say) Telstra cable or ADSL without a fixed-IP address and not using the Telstra mail server (which is very common), will get rejected.

10.7 Configuring

- FEATURE(‘dnsbl’)
- FEATURE(‘dnsbl’,’relays.ordb.org’)
- FEATURE(‘dnsbl’,’opm.blitzed.org’, ‘”451 Temporarily rejected from proxy list”’)

Notes. . .

The first one is the simplest, but requires that you have a subscription to MAPS (`mail-abuse.org`).

The second one is what you will use most of the time – including the appropriate DNS domain to look up. This will create a rejection message like this:

```
Mail from IP-ADDRESS refused by blackhole site SERVER
```

The third example customizes the message (including making it temporary rather than permanent).

10.8 More information

The whole scoop from `cf/README`

Notes. . .

```
The primary anti-spam features available in sendmail are:
* Relaying is denied by default.
* Better checking on sender information.
* Access database.
* Header checks

Relaying (transmission of messages from a site outside your host (class
{w}) to another site except yours) is denied by default.  Note that this
changed in sendmail 8.9; previous versions allowed relaying by default.
If you really want to revert to the old behaviour, you will need to use
FEATURE(‘promiscuous_relay’).  You can allow certain domains to relay
through your server by adding their domain name or IP address to class
{R} using RELAY_DOMAIN() and RELAY_DOMAIN_FILE() or via the access database
(described below).  Note that IPv6 addresses must be prefaced with "IPv6:".
The file consists (like any other file based class) of entries listed on
separate lines, e.g.,

        sendmail.org
        128.32
        IPv6:2002:c0a8:02c7
        IPv6:2002:c0a8:51d2::23f4
        host.mydomain.com
        [UNIX:localhost]

Notice: the last entry allows relaying for connections via a UNIX
socket to the MTA/MSP.  This might be necessary if your configuration
doesn’t allow relaying by other means in that case, e.g., by having
localhost.$m in class {R} (make sure $m is not just a top level
domain).

If you use

        FEATURE(‘relay_entire_domain’)}

then any host in any of your local domains (that is, class {{m}})
will be relayed (that is, you will accept mail either to or from any
host in your domain).

You can also allow relaying based on the MX records of the host
portion of an incoming recipient address by using
```

```
        FEATURE(‘relay_based_on_MX’)
```

For example, if your server receives a recipient of user@domain.com and domain.com lists your server in its MX records, the mail will be accepted for relay to domain.com. This feature may cause problems if MX lookups for the recipient domain are slow or time out. In that case, mail will be temporarily rejected. It is usually better to maintain a list of hosts/domains for which the server acts as relay. Note also that this feature will stop spammers from using your host to relay spam but it will not stop outsiders from using your server as a relay for their site (that is, they set up an MX record pointing to your mail server, and you will relay mail addressed to them without any prior arrangement). Along the same lines,

```
        FEATURE(‘relay_local_from’)
```

will allow relaying if the sender specifies a return path (i.e. MAIL FROM: <user@domain>) domain which is a local domain. This is a dangerous feature as it will allow spammers to spam using your mail server by simply specifying a return address of user@your.domain.com. It should not be used unless absolutely necessary.
A slightly better solution is

```
        FEATURE(‘relay_mail_from’)
```

which allows relaying if the mail sender is listed as RELAY in the access map. If an optional argument ‘domain’ is given, the domain portion of the mail sender is also checked to allowing relaying. This option only works together with the tag From: for the LHS of the access map entries (see below: Finer control...). This feature allows spammers to abuse your mail server by specifying a return address that you enabled in your access file. This may be harder to figure out for spammers, but it should not be used unless necessary. Instead use SMTP AUTH or STARTTLS to allow relaying for roaming users.

If source routing is used in the recipient address (e.g., RCPT TO: <user%site.com@othersite.com>), sendmail will check user@site.com for relaying if othersite.com is an allowed relay host in either class {R}, class {m} if FEATURE(‘relay_entire_domain’) is used, or the access database if FEATURE(‘access_db’) is used. To prevent the address from being stripped down, use:

```
        FEATURE(‘loose_relay_check’)
```

If you think you need to use this feature, you probably do not. This should only be used for sites which have no control over the addresses that they provide a gateway for. Use this FEATURE with caution as it can allow spammers to relay through your server if not setup properly.

NOTICE: It is possible to relay mail through a system which the anti-relay rules do not prevent: the case of a system that does use FEATURE(‘nouucp’, ‘nospecial’) (system A) and relays local messages to a mail hub (e.g., via LOCAL_RELAY or LUSER_RELAY) (system B). If system B doesn’t use FEATURE(‘nouucp’) at all, addresses of the form <example.net!user@local.host> would be relayed to <user@example.net>. System A doesn’t recognize ‘!’ as an address separator and therefore forwards it to the mail hub which in turns relays it because it came from a trusted local host. So if a mailserver allows UUCP (bang-format) addresses, all systems from which it allows relaying should do the same or reject those addresses.

As of 8.9, sendmail will refuse mail if the MAIL FROM: parameter has an unresolvable domain (i.e., one that DNS, your local name service, or special case rules in ruleset 3 cannot locate). This also applies to addresses that use domain literals, e.g., <user@[1.2.3.4]>, if the IP address can’t be mapped to a host name. If you want to continue to accept such domains, e.g., because you are inside a firewall that has only a limited view of the Internet host name space (note that you will not be able to return mail to them unless you have some "smart host" forwarder), use

```
        FEATURE(‘accept_unresolvable_domains’)
```

Alternatively, you can allow specific addresses by adding them to the access map, e.g.,

```
        From:unresolvable.domain        OK
        From:[1.2.3.4]                  OK
        From:[1.2.4]                    OK
```

Notice: domains which are temporarily unresolvable are (temporarily) rejected with a 451 reply code. If those domains should be accepted (which is discouraged) then you can use

```
        LOCAL_CONFIG
        C{ResOk}TEMP
```

sendmail will also refuse mail if the MAIL FROM: parameter is not fully qualified (i.e., contains a domain as well as a user). If you want to continue to accept such senders, use

```
        FEATURE(‘accept_unqualified_senders’)
```

Setting the DaemonPortOptions modifier ’u’ overrides the default behavior, i.e., unqualified addresses are accepted even without this FEATURE. If this FEATURE is not used, the DaemonPortOptions modifier ’f’ can be used to enforce fully qualified domain names.

An ‘‘access’’ database can be created to accept or reject mail from selected domains. For example, you may choose to reject all mail originating from known spammers. To enable such a database, use

```
        FEATURE(‘access_db’)
```

Notice: the access database is applied to the envelope addresses and the connection information, not to the header.

The FEATURE macro can accept as second parameter the key file definition for the database; for example

```
        FEATURE(‘access_db’, ‘hash -T<TMPF> /etc/mail/access_map’)
```

Notice: If a second argument is specified it must contain the option ‘-T<TMPF>’ as shown above. The optional third and fourth parameters may be ‘skip’ or ‘lookupdotdomain’. The former enables SKIP as value part (see below), the latter is another way to enable the feature of the same name (see above).

Remember, since /etc/mail/access is a database, after creating the text file as described below, you must use makemap to create the database map. For example:

```
        makemap hash /etc/mail/access < /etc/mail/access
```

The table itself uses e-mail addresses, domain names, and network numbers as keys. Note that IPv6 addresses must be prefaced with "IPv6:". For example,

```
        spammer@aol.com                 REJECT
        cyberspammer.com                REJECT
        192.168.212                     REJECT
        IPv6:2002:c0a8:02c7             RELAY
        IPv6:2002:c0a8:51d2::23f4       REJECT
```

would refuse mail from spammer@aol.com, any user from cyberspammer.com (or any host within the cyberspammer.com domain), any host on the 192.168.212.* network, and the IPv6 address 2002:c0a8:51d2::23f4. It would allow relay for the IPv6 network 2002:c0a8:02c7::/48.

The value part of the map can contain:

```
        OK              Accept mail even if other rules in the running
                        ruleset would reject it, for example, if the domain
                        name is unresolvable.  "Accept" does not mean
                        "relay", but at most acceptance for local
                        recipients.  That is, OK allows less than RELAY.
        RELAY           Accept mail addressed to the indicated domain or
```

```
                        received from the indicated domain for relaying
                        through your SMTP server.  RELAY also serves as
                        an implicit OK for the other checks.
        REJECT          Reject the sender or recipient with a general
                        purpose message.
        DISCARD         Discard the message completely using the
                        $#discard mailer.  If it is used in check_compat,
                        it affects only the designated recipient, not
                        the whole message as it does in all other cases.
                        This should only be used if really necessary.
        SKIP            This can only be used for host/domain names
                        and IP addresses/nets.  It will abort the current
                        search for this entry without accepting or rejecting
                        it but causing the default action.
        ### any text    where ### is an RFC 821 compliant error code and
                        "any text" is a message to return for the command.
                        The string should be quoted to avoid surprises,
                        e.g., sendmail may remove spaces otherwise.
                        This type is deprecated, use one the two
                        ERROR:  entries below instead.
        ERROR:### any text
                        as above, but useful to mark error messages as such.
        ERROR:D.S.N:### any text
                        where D.S.N is an RFC 1893 compliant error code
                        and the rest as above.
```

For example:

```
        cyberspammer.com        ERROR:550 "We don't accept mail from spammers"
        okay.cyberspammer.com   OK
        sendmail.org            RELAY
        128.32                  RELAY
        IPv6:1:2:3:4:5:6:7      RELAY
        [127.0.0.3]             OK
        [IPv6:1:2:3:4:5:6:7:8]  OK
```

would accept mail from okay.cyberspammer.com, but would reject mail from all other hosts at cyberspammer.com with the indicated message. It would allow relaying mail from and to any hosts in the sendmail.org domain, and allow relaying from the 128.32.*.* network and the IPv6 1:2:3:4:5:6:7:* network. The latter two entries are for checks against ${client_name} if the IP address doesn't resolve to a hostname (or is considered as "may be forged"). That is, using square brackets means these are host names, not network numbers.

Warning: if you change the RFC 821 compliant error code from the default value of 550, then you should probably also change the RFC 1893 compliant error code to match it. For example, if you use

```
        user@example.com        ERROR:450 mailbox full
```

the error returned would be "450 5.0.0 mailbox full" which is wrong. Use "ERROR:4.2.2:450 mailbox full" instead.

Note, UUCP users may need to add hostname.UUCP to the access database or class {R}.

If you also use:

```
        FEATURE(‘relay_hosts_only’)
```

then the above example will allow relaying for sendmail.org, but not hosts within the sendmail.org domain. Note that this will also require hosts listed in class {R} to be fully qualified host names.

You can also use the access database to block sender addresses based on the username portion of the address. For example:

```
        FREE.STEALTH.MAILER@    ERROR:550 Spam not accepted
```

Note that you must include the @ after the username to signify that this database entry is for checking only the username portion of the sender address.

If you use:

```
        FEATURE(‘blacklist_recipients’)
```

then you can add entries to the map for local users, hosts in your domains, or addresses in your domain which should not receive mail:

```
        badlocaluser@           ERROR:550 Mailbox disabled for this username
        host.mydomain.com       ERROR:550 That host does not accept mail
        user@otherhost.mydomain.com     ERROR:550 Mailbox disabled for this recipient
```

This would prevent a recipient of badlocaluser@mydomain.com, any user at host.mydomain.com, and the single address user@otherhost.mydomain.com from receiving mail. Please note: a local username must be now tagged with an @ (this is consistent with the check of the sender address, and hence it is possible to distinguish between hostnames and usernames). Enabling this feature will keep you from sending mails to all addresses that have an error message or REJECT as value part in the access map. Taking the example from above:

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

Mail can’t be sent to spammer@aol.com or anyone at cyberspammer.com.

There are several DNS based blacklists, the first of which was the RBL (‘‘Realtime Blackhole List’’) run by the MAPS project, see http://mail-abuse.org/. These are databases of spammers maintained in DNS. To use such a database, specify

```
        FEATURE(‘dnsbl’)
```

This will cause sendmail to reject mail from any site in the original Realtime Blackhole List database. This default DNS blacklist, blackholes.mail-abuse.org, is a service offered by the Mail Abuse Prevention System (MAPS). As of July 31, 2001, MAPS is a subscription service, so using that network address won’t work if you haven’t subscribed. Contact MAPS to subscribe (http://mail-abuse.org/).

You can specify an alternative RBL server to check by specifying an argument to the FEATURE. The default error message is

```
        Mail from IP-ADDRESS refused by blackhole site SERVER
```

where IP-ADDRESS and SERVER are replaced by the appropriate information. A second argument can be used to specify a different text. By default, temporary lookup failures are ignored and hence cause the connection not to be rejected by the DNS based rejection list. This behavior can be changed by specifying a third argument, which must be either ‘t’ or a full error message. For example:

```
        FEATURE(‘dnsbl’, ‘dnsbl.example.com’, ‘’,
        ‘"451 Temporary lookup failure for " $&{client_addr} " in dnsbl.example.com"’
```

If ‘t’ is used, the error message is:

```
        451 Temporary lookup failure of IP-ADDRESS at SERVER
```

where IP-ADDRESS and SERVER are replaced by the appropriate information.

This FEATURE can be included several times to query different DNS based rejection lists, e.g., the dial-up user list (see http://mail-abuse.org/dul/).

Notice: to avoid checking your own local domains against those blacklists, use the access_db feature and add:

```
        Connect:10.1            OK
        Connect:127.0.0.1       RELAY
```

to the access map, where 10.1 is your local network. You may want to use "RELAY" instead of "OK" to allow also relaying instead of just disabling the DNS lookups in the backlists.

The features described above make use of the check_relay, check_mail, and check_rcpt rulesets. If you wish to include your own checks, you can put your checks in the rulesets Local_check_relay, Local_check_mail, and Local_check_rcpt. For example if you wanted to block senders with all numeric usernames (i.e. 2312343@bigisp.com), you would use Local_check_mail and the regex map:

```
        LOCAL_CONFIG
        Kallnumbers regex -a@MATCH ^[0-9]+$

        LOCAL_RULESETS
        SLocal_check_mail
        # check address against various regex checks
        R$*                             $: $>Parse0 $>3 $1
        R$+ < @ bigisp.com. > $*        $: $(allnumbers $1 $)
        R@MATCH                         $#error $: 553 Header Error
```

These rules are called with the original arguments of the corresponding check_* ruleset. If the local ruleset returns $#OK, no further checking is done by the features described above and the mail is accepted. If the local ruleset resolves to a mailer (such as $#error or $#discard), the appropriate action is taken. Otherwise, the results of the local rewriting are ignored.

Finer control by using tags for the LHS of the access map

Read this section only if the options listed so far are not sufficient for your purposes. There is now the option to tag entries in the access map according to their type. Three tags are available:

Connect:	connection information (${client_addr}, ${client_name})
From:	envelope sender
To:	envelope recipient

If the required item is looked up in a map, it will be tried first with the corresponding tag in front, then (as fallback to enable backward compatibility) without any tag, unless the specific feature requires a tag. For example,

```
        From:spammer@some.dom   REJECT
        To:friend.domain        RELAY
        Connect:friend.domain   OK
        Connect:from.domain     RELAY
        From:good@another.dom   OK
        From:another.dom        REJECT
```

This would deny mails from spammer@some.dom but you could still send mail to that address even if FEATURE(‘blacklist_recipients’) is enabled. Your system will allow relaying to friend.domain, but not from it (unless enabled by other means). Connections from that domain will be allowed even if it ends up in one of the DNS based rejection lists. Relaying is enabled from from.domain but not to it (since relaying is based on the connection information for outgoing relaying, the tag Connect: must be used; for incoming relaying, which is based on the recipient address, To: must be used). The last two entries allow mails from good@another.dom but reject mail from all other addresses with another.dom as domain part.

Delay all checks

By using FEATURE(‘delay_checks’) the rulesets check_mail and check_relay will not be called when a client connects or issues a MAIL command, respectively. Instead, those rulesets will be called by the check_rcpt ruleset; they will be skipped if a sender has been authenticated using a "trusted" mechanism, i.e., one that is defined via TRUST_AUTH_MECH(). If check_mail returns an error then the RCPT TO command will be rejected with that error. If it returns some other result starting with $# then check_relay will be skipped. If the sender address (or a part of it) is listed in the access map and it has a RHS of OK or RELAY, then check_relay will be skipped. This has an interesting side effect: if your domain is my.domain and you have

```
        my.domain       RELAY
```

in the access map, then all e-mail with a sender address of <user@my.domain> gets through, even if check_relay would reject it (e.g., based on the hostname or IP address). This allows spammers to get around DNS based blacklist by faking the sender address. To avoid this problem you have to use tagged entries:

```
        To:my.domain            RELAY
        Connect:my.domain       RELAY
```

if you need those entries at all (class {R} may take care of them).

FEATURE(‘delay_checks’) can take an optional argument:

```
        FEATURE(‘delay_checks’, ‘friend’)
                enables spamfriend test
        FEATURE(‘delay_checks’, ‘hater’)
                enables spamhater test
```

If such an argument is given, the recipient will be looked up in the access map (using the tag Spam:). If the argument is ‘friend’, then the other rulesets will be skipped if the recipient address is found and has RHS friend. If the argument is ‘hater’, then the other rulesets will be applied if the recipient address is found and has RHS hater.

This allows for simple exceptions from the tests, e.g., by activating the friend option and having

```
        Spam:abuse@     FRIEND
```

in the access map, mail to abuse@localdomain will get through. It is also possible to specify a full address or an address with +detail:

```
        Spam:abuse@my.domain    FRIEND
        Spam:me+abuse@          FRIEND
        Spam:spam.domain        FRIEND
```

Note: The required tag has been changed in 8.12 from To: to Spam:. This change is incompatible to previous versions. However, you can (for now) simply add the new entries to the access map, the old ones will be ignored. As soon as you removed the old entries from the access map, specify a third parameter ('n') to this feature and the backward compatibility rules will not be in the generated .cf file.

Header Checks

You can also reject mail on the basis of the contents of headers. This is done by adding a ruleset call to the 'H' header definition command in sendmail.cf. For example, this can be used to check the validity of a Message-ID: header:

```
        LOCAL_RULESETS
        HMessage-Id: $>CheckMessageId

        SCheckMessageId
        R< $+ @ $+ >            $@ OK
        R$*                     $#error $: 553 Header Error
```

The alternative format:

```
        HSubject: $>+CheckSubject
```

that is, $>+ instead of $>, gives the full Subject: header including comments to the ruleset (comments in parentheses () are stripped by default).

A default ruleset for headers which don't have a specific ruleset defined for them can be given by:

```
        H*: $>CheckHdr
```

Notice:
1. All rules act on tokens as explained in doc/op/op.{me,ps,txt}. That may cause problems with simple header checks due to the tokenization. It might be simpler to use a regex map and apply it to $&{currHeader}.
2. There are no default rulesets coming with this distribution of sendmail. You can either write your own or you can search the WWW for examples, e.g., http://www.digitalanswers.org/check_local/

After all of the headers are read, the check_eoh ruleset will be called for any final header-related checks. The ruleset is called with the number of headers and the size of all of the headers in bytes separated by $|. One example usage is to reject messages which do not have a Message-Id: header. However, the Message-Id: header is *NOT* a required header and is not a guaranteed spam indicator. This ruleset is an example and should probably not be used in production.

```
        LOCAL_CONFIG
        Kstorage macro

        LOCAL_RULESETS
        HMessage-Id: $>CheckMessageId

        SCheckMessageId
        # Record the presence of the header
        R$*                     $: $(storage {MessageIdCheck} $@ OK $) $1
        R< $+ @ $+ >            $@ OK
        R$*                     $#error $: 553 Header Error

        Scheck_eoh
        # Check the macro
        R$*                     $: < $&{MessageIdCheck} >
        # Clear the macro for the next message
        R$*                     $: $(storage {MessageIdCheck} $) $1
        # Has a Message-Id: header
        R< $+ >                 $@ OK
        # Allow missing Message-Id: from local mail
        R$*                     $: < $&{client_name} >
        R< >                    $@ OK
        R< $=w >                $@ OK
        # Otherwise, reject the mail
        R$*                     $#error $: 553 Header Error
```

10.9 Anti-spam exercises

Using a blacklist

Notes. . .

- If you can do so, create a new zone in your site's DNS for blacklist testing, do so now. Otherwise, your instructor will come up with something equivalent.
- Rebuild your `sendmail.cf` with a few DNS blacklists as `dnsbl` arguments, including the test one from step 1 if possible.
- Test sending a message from a blacklisted IP address. What do you get back?

Chapter 11

Integration with other services

11.1 IMAP, POP – Plan 1

- *sendmail* delivers into `/var/mail` files or equivalent
- POP/IMAP server needs to read from there
- Both WU-IMAP and Dovecot can do this
- Run `imapd` and/or `pop3d` from `inetd`

Notes. . .

WU-IMAP can be found at `www.washington.edu/imap`. Documentation for how to set it up is included – it's very much compile and run.

Dovecot is a small, simple and secure IMAP server, developed by Procontrol – `dovecot.procontrol.fi`.

Newer Linux systems are moving to `xinetd` rather than `inetd`. The only difference really for an administrator is that instead of modifying `/etc/inetd.conf`, you will be creating or modifying `/etc/xinetd.d/imapd`.

11.2 Plan 1 problems

- `imapd` has to re-read and re-write the user's whole mail file.
- No support for folders
- Still OK for `pop3d` though.

Notes...

A user's `/var/mail` file could be huge. As long as they leave their IMAP connection open, this isn't a big problem; but if they constantly disconnect and reconnect, the overhead on the server is considerable. Web-based mail readers do a lot of this because of the stateless nature of the web.

11.3 IMAP, POP – Plan 2

- Deliver via `procmail` into a maildir
- Use `courier-imap` and `courier-pop3d`

Notes. . .

Maildir format was initially developed for `qmail` and then adopted by `postfix`. It stores each mail message in a separate file in a sub-directory of the user's home directory.

If your user's home directories are shared via NFS, then the IMAP server can run on a different system to the mail server.

The `courier-imap` and `courier-pop3d` servers are part of the `courier-mta` suite, which is a complete *sendmail* replacement, but along with their web-mail server and mail filtering system, these components can be used separately. Everything you want to know is at `www.courier-mta.org`.

11.4 Web-based mail

- `IMP` or many alternatives.
- Talks to IMAP server

Notes. . .

IMP is part of the Horde suite and can be found at `www.horde.org/imp`. It is quite impressive in its functionality and user-interface.

It does not need to run on the same server as the IMAP service – indeed, it can be configured to allow users to decide which IMAP services they wish to connect to.

11.5 Exercises

Just a quick demonstration of integration...

Notes...

1. OpenBSD includes a copy of `popa3d` by Solar Designer. It is not enabled by default. Edit `/etc/inetd.conf` and then signal `inetd` to restart (send it a HUP signal). If you are not using OpenBSD, ask your instructor for the equivalent on your platform.
2. Configure a mail client (e.g. MS Outlook, Eudora, Mozilla, etc.) to ask a POP3 mail account on your server. Use an ordinary user account (and password).
3. Send a mail message to that ordinary user, and then fetch the mail on your client.

Chapter 12

Milters

12.1 What is a milter?

- A filter for mail
- Every message gets sent through all milters
- Can do virus or spam checks
- Connected to *sendmail* via sockets

Notes. . .

The sockets can either be TCP/IP or Unix-domain sockets.

12.2 How to use milters

- `InputMailFilters` gives order
- `X` configuration line

Notes. . .

Here is a sample excerpt from a `sendmail.cf` with a milter:

```
Xvirscan, S=inet:3333@localhost, F=T
```

There are just three possible arguments (which as usual can be reduced to single letters):

1. Socket (what socket to talk to the milter on)
2. Flags (any special options for this milter)
3. Timeouts

The possible flags are just for how to handle an unavailable milter:

R Reject connection if filter unavailable.

T Temporary fail connection if filter unavailable.

Chapter 13

Appendix: LDAP

From the `cf/README` file:

13.1 Using LDAP for aliases, maps and classes

LDAP can be used for aliases, maps, and classes by either specifying your own LDAP map specification or using the built-in default LDAP map specification. The built-in default specifications all provide lookups which match against either the machine's fully qualified hostname (`$j`) or a "cluster". The cluster allows you to share LDAP entries among a large number of machines without having to enter each of the machine names into each LDAP entry. To set the LDAP cluster name to use for a particular machine or set of machines, set the confLDAP_CLUSTER m4 variable to a unique name. For example:

```
define(‘confLDAP_CLUSTER’, ‘Servers’)
```

Here, the word ‘Servers’ will be the cluster name. As an example, assume that smtp.sendmail.org, etrn.sendmail.org, and mx.sendmail.org all belong to the Servers cluster.

Some of the LDAP LDIF examples below show use of the Servers cluster. Every entry must have either a sendmailMTAHost or sendmailMTACluster attribute or it will be ignored. Be careful as mixing clusters and individual host records can have surprising results (see the CAUTION sections below).

See the file cf/sendmail.schema for the actual LDAP schemas. Note that this schema (and therefore the lookups and examples below) is experimental at this point as it has had little public review. Therefore, it may change in future versions. Feedback via sendmail@sendmail.org is encouraged.

13.1.1 Aliases

The ALIAS_FILE (O AliasFile) option can be set to use LDAP for alias lookups. To use the default schema, simply use:

```
define(‘ALIAS_FILE’, ‘ldap:’)
```

By doing so, you will use the default schema which expands to a map declared as follows:

```
ldap -k (&(objectClass=sendmailMTAAliasObject)
  (sendmailMTAAliasGrouping=aliases)
  (|(sendmailMTACluster=${sendmailMTACluster})
    (sendmailMTAHost=$j))
  (sendmailMTAKey=%0))
     -v sendmailMTAAliasValue
```

NOTE: The macros shown above `$sendmailMTACluster` and `$j` are not actually used when the binary expands the ‘ldap:’ token as the AliasFile option is not actually macro-expanded when read from the `sendmail.cf` file.

Example LDAP LDIF entries might be:

```
dn: sendmailMTAKey=sendmail-list, dc=sendmail, dc=org
objectClass: sendmailMTA
objectClass: sendmailMTAAlias
objectClass: sendmailMTAAliasObject
sendmailMTAAliasGrouping: aliases
sendmailMTAHost: etrn.sendmail.org
sendmailMTAKey: sendmail-list
sendmailMTAAliasValue: ca@example.org
sendmailMTAAliasValue: eric
sendmailMTAAliasValue: gshapiro@example.com

dn: sendmailMTAKey=owner-sendmail-list, dc=sendmail, dc=org
objectClass: sendmailMTA
objectClass: sendmailMTAAlias
objectClass: sendmailMTAAliasObject
sendmailMTAAliasGrouping: aliases
sendmailMTAHost: etrn.sendmail.org
sendmailMTAKey: owner-sendmail-list
sendmailMTAAliasValue: eric

dn: sendmailMTAKey=postmaster, dc=sendmail, dc=org
objectClass: sendmailMTA
objectClass: sendmailMTAAlias
objectClass: sendmailMTAAliasObject
sendmailMTAAliasGrouping: aliases
```

```
sendmailMTACluster: Servers
sendmailMTAKey: postmaster
sendmailMTAAliasValue: eric
```

Here, the aliases sendmail-list and owner-sendmail-list will be available only on etrn.sendmail.org but the postmaster alias will be available on every machine in the Servers cluster (including etrn.sendmail.org).

CAUTION: aliases are additive so that entries like these:

```
dn: sendmailMTAKey=bob, dc=sendmail, dc=org
objectClass: sendmailMTA
objectClass: sendmailMTAAlias
objectClass: sendmailMTAAliasObject
sendmailMTAAliasGrouping: aliases
sendmailMTACluster: Servers
sendmailMTAKey: bob
sendmailMTAAliasValue: eric

dn: sendmailMTAKey=bob, dc=sendmail, dc=org
objectClass: sendmailMTA
objectClass: sendmailMTAAlias
objectClass: sendmailMTAAliasObject
sendmailMTAAliasGrouping: aliases
sendmailMTAHost: etrn.sendmail.org
sendmailMTAKey: bob
sendmailMTAAliasValue: gshapiro
```

would mean that on all of the hosts in the cluster, mail to bob would go to eric EXCEPT on etrn.sendmail.org in which case it would go to BOTH eric and gshapiro.

If you prefer not to use the default LDAP schema for your aliases, you can specify the map parameters when setting ALIAS_FILE. For example:

```
define('ALIAS_FILE', 'ldap:-k (&(objectClass=mailGroup)(mail=%0)) -v
mgrpRFC822MailMember')
```

13.1.2 Maps

FEATURE()'s which take an optional map definition argument (e.g., access, mailertable, virtusertable, etc.) can instead take the special keyword 'LDAP', e.g.:

- FEATURE('access_db', 'LDAP')
- FEATURE('virtusertable', 'LDAP')

When this keyword is given, that map will use LDAP lookups consisting of the objectClass sendmailMTAClassObject, the attribute sendmailMTAMapName with the map name, a search attribute of sendmailMTAKey, and the value attribute sendmailMTAMapValue.

The values for sendmailMTAMapName are:

FEATURE()	sendmailMTAMapName
access_db	access
authinfo	authinfo
bitdomain	bitdomain
domaintable	domain
genericstable	generics
mailertable	mailer
uucpdomain	uucpdomain
virtusertable	virtuser

For example, FEATURE('mailertable', 'LDAP') would use the map definition:

```
Kmailertable ldap -k (&(objectClass=sendmailMTAMapObject)
       (sendmailMTAMapName=mailer)
       (|(sendmailMTACluster=${sendmailMTACluster})
 (sendmailMTAHost=$j))
       (sendmailMTAKey=%0))
  -1 -v sendmailMTAMapValue
```

An example LDAP LDIF entry using this map might be:

```
dn: sendmailMTAMapName=mailer, dc=sendmail, dc=org
objectClass: sendmailMTA
objectClass: sendmailMTAMap
sendmailMTACluster: Servers
sendmailMTAMapName: mailer

dn: sendmailMTAKey=example.com, sendmailMTAMapName=mailer, dc=sendmail, dc=org
objectClass: sendmailMTA
objectClass: sendmailMTAMap
objectClass: sendmailMTAMapObject
sendmailMTAMapName: mailer
sendmailMTACluster: Servers
sendmailMTAKey: example.com
sendmailMTAMapValue: relay:[smtp.example.com]
```

CAUTION: If your LDAP database contains the record above and *ALSO* a host specific record such as:

```
dn: sendmailMTAKey=example.com@etrn, sendmailMTAMapName=mailer, dc=sendmail, dc=org
objectClass: sendmailMTA
objectClass: sendmailMTAMap
```

```
objectClass: sendmailMTAMapObject
sendmailMTAMapName: mailer
sendmailMTAHost: etrn.sendmail.org
sendmailMTAKey: example.com
sendmailMTAMapValue: relay:[mx.example.com]
```

then these entries will give unexpected results. When the lookup is done on etrn.sendmail.org, the effect is that there is *NO* match at all as maps require a single match. Since the host etrn.sendmail.org is also in the Servers cluster, LDAP would return two answers for the example.com map key in which case sendmail would treat this as no match at all.

If you prefer not to use the default LDAP schema for your maps, you can specify the map parameters when using the FEATURE(). For example:

```
FEATURE(‘access_db’, ‘ldap:-1 -k (&(objectClass=mapDatabase)(key=%0))
-v value’)
```

13.1.3 Classes

Normally, classes can be filled via files or programs. As of 8.12, they can also be filled via map lookups using a new syntax:

```
FClassNamemapkey@mapclass:mapspec
```

mapkey is optional and if not provided the map key will be empty. This can be used with LDAP to read classes from LDAP. Note that the lookup is only done when sendmail is initially started. Use the special value ‘@LDAP’ to use the default LDAP schema. For example:

```
RELAY_DOMAIN_FILE(‘@LDAP’)
```

would put all of the attribute sendmailMTAClassValue values of LDAP records with objectClass sendmailMTAClass and an attribute sendmailMTAClassName of ’R’ into class `$=R`. In other words, it is equivalent to the LDAP map specification:

```
F{R}@ldap:-k (&(objectClass=sendmailMTAClass)
       (sendmailMTAClassName=R)
       (|(sendmailMTACluster=${sendmailMTACluster})
 (sendmailMTAHost=$j)))
  -v sendmailMTAClassValue
```

NOTE: The macros shown above `$sendmailMTACluster` and `$j` are not actually used when the binary expands the ‘@LDAP’ token as class declarations are not actually macro-expanded when read from the sendmail.cf file.

This can be used with class related commands such as RELAY_DOMAIN_FILE(), MASQUERADE_DOMAIN_FILE(), etc:

Command	sendmailMTAClassName
CANONIFY_DOMAIN_FILE()	Canonify
EXPOSED_USER_FILE()	E
GENERICS_DOMAIN_FILE()	G
LDAPROUTE_DOMAIN_FILE()	LDAPRoute
LDAPROUTE_EQUIVALENT_FILE()	LDAPRouteEquiv
LOCAL_USER_FILE()	L
MASQUERADE_DOMAIN_FILE()	M
MASQUERADE_EXCEPTION_FILE()	N
RELAY_DOMAIN_FILE()	R
VIRTUSER_DOMAIN_FILE()	VirtHost

You can also add your own as any ’F’ile class of the form:

```
F{ClassName}@LDAP
  ^^^^^^^^^
```

will use ”ClassName” for the sendmailMTAClassName.

An example LDAP LDIF entry would look like:

```
dn: sendmailMTAClassName=R, dc=sendmail, dc=org
objectClass: sendmailMTA
objectClass: sendmailMTAClass
sendmailMTACluster: Servers
sendmailMTAClassName: R
sendmailMTAClassValue: sendmail.org
sendmailMTAClassValue: example.com
sendmailMTAClassValue: 10.56.23
```

CAUTION: If your LDAP database contains the record above and *ALSO* a host specific record such as:

```
dn: sendmailMTAClassName=R@etrn.sendmail.org, dc=sendmail, dc=org
objectClass: sendmailMTA
objectClass: sendmailMTAClass
sendmailMTAHost: etrn.sendmail.org
sendmailMTAClassName: R
sendmailMTAClassValue: example.com
```

the result will be similar to the aliases caution above. When the lookup is done on etrn.sendmail.org, `$=R` would contain all of the entries (from both the cluster match and the host match). In other words, the effective is additive.

If you prefer not to use the default LDAP schema for your classes, you can specify the map parameters when using the class command. For example:

```
VIRTUSER_DOMAIN_FILE(‘@ldap:-k (&(objectClass=virtHosts)(host=*)) -v host’)
```

Remember, macros can not be used in a class declaration as the binary does not expand them.

13.2 LDAP Routing

FEATURE(‘ldap_routing’) can be used to implement the IETF Internet Draft LDAP Schema for Intranet Mail Routing (draft-lachman-laser-ldap-mail-routing-01). This feature enables LDAP-based rerouting of a particular address to either a different host or a different address. The LDAP lookup is first attempted on the full address (e.g., user@example.com) and then on the domain portion (e.g., @example.com). Be sure to setup your domain for LDAP routing using LDAPROUTE_DOMAIN(), e.g.:

```
LDAPROUTE_DOMAIN(‘example.com’)
```

Additionally, you can specify equivalent domains for LDAP routing using LDAPROUTE_EQUIVALENT() and LDAPROUTE_EQUIVALENT_FILE(). ’Equivalent’ hostnames are mapped to `$M` (the masqueraded hostname for the server) before the LDAP query. For example, if the mail is addressed to user@host1.example.com, normally the LDAP lookup would only be done for ’user@host1.example.com’ and ’@host1.example.com’. However, if LDAPROUTE_EQUIVALENT(‘host1.example.com’) is used, the lookups would also be done on ’user@example.com’ and ’@example.com’ after attempting the host1.example.com lookups.

By default, the feature will use the schemas as specified in the draft and will not reject addresses not found by the LDAP lookup. However, this behavior can be changed by giving additional arguments to the FEATURE() command:

```
 FEATURE(‘ldap_routing’, <mailHost>, <mailRoutingAddress>, <bounce>, <detail>)
```

where `<mailHost>` is a map definition describing how to lookup an alternative mail host for a particular address; ¡mailRoutingAddress¿ is a map definition describing how to lookup an alternative address for a particular address; the `<bounce>` argument, if present and not the word ”passthru”, dictates that mail should be bounced if neither a mailHost nor mailRoutingAddress is found; and `<detail>` indicates what actions to take if the address contains +detail information – ‘strip’ tries the lookup with the +detail and if no matches are found, strips the +detail and tries the lookup again; ‘preserve’, does the same as ‘strip’ but if a mailRoutingAddress match is found, the +detail information is copied to the new address.

The default `<mailHost>` map definition is:

```
ldap -1 -v mailHost -k (&(objectClass=inetLocalMailRecipient) (mailLocalAddress=%0))
```

The default `<mailRoutingAddress>` map definition is:

```
ldap -1 -v mailRoutingAddress -k (&(objectClass=inetLocalMailRecipient)
(mailLocalAddress=%0))
```

Note that neither includes the LDAP server hostname (-h server) or base DN (-b o=org,c=COUNTRY), both necessary for LDAP queries. It is presumed that your .mc file contains a setting for the confLDAP_DEFAULT_SPEC option with these settings. If this is not the case, the map definitions should be changed as described above.

The following possibilities exist as a result of an LDAP lookup on an address:

mailHost is	mailRoutingAddress is	Results in
set to a "local" host	set	mail delivered to mailRoutingAddress
set to a "local" host	not set	delivered to original address
set to a remote host	set	mailRoutingAddress relayed to mailHost
set to a remote host	not set	original address relayed to mailHost
not set	set	mail delivered to mailRoutingAddress
not set	not set	delivered to original address *OR* bounced as unknown user

The term "local" host above means the host specified is in class `w`. If the result would mean sending the mail to a different host, that host is looked up in the mailertable before delivery.

Note that the last case depends on whether the third argument is given to the FEATURE() command. The default is to deliver the message to the original address.

The LDAP entries should be set up with an objectClass of inetLocalMailRecipient and the address be listed in a mailLocalAddress attribute. If present, there must be only one mailHost attribute and it must contain a fully qualified host name as its value. Similarly, if present, there must be only one mailRoutingAddress attribute and it must contain an RFC 822 compliant address. Some example LDAP records (in LDIF format):

```
dn: uid=tom, o=example.com, c=US
objectClass: inetLocalMailRecipient
mailLocalAddress: tom@example.com
mailRoutingAddress: thomas@mailhost.example.com
```

This would deliver mail for tom@example.com to thomas@mailhost.example.com.

```
dn: uid=dick, o=example.com, c=US
objectClass: inetLocalMailRecipient
mailLocalAddress: dick@example.com
mailHost: eng.example.com
```

This would relay mail for dick@example.com to the same address but redirect the mail to MX records listed for the host eng.example.com (unless the mailertable overrides).

```
dn: uid=harry, o=example.com, c=US
objectClass: inetLocalMailRecipient
mailLocalAddress: harry@example.com
mailHost: mktmail.example.com
mailRoutingAddress: harry@mkt.example.com
```

This would relay mail for harry@example.com to the MX records listed for the host mktmail.example.com using the new address harry@mkt.example.com when talking to that host.

```
dn: uid=virtual.example.com, o=example.com, c=US
objectClass: inetLocalMailRecipient
mailLocalAddress: @virtual.example.com
mailHost: server.example.com
mailRoutingAddress: virtual@example.com
```

This would send all mail destined for any username @virtual.example.com to the machine server.example.com's MX servers and deliver to the address virtual@example.com on that relay machine.

Chapter 14

Appendix: All Macros and Options

14.1 Macros

$a The origination date in RFC 822 format. This is extracted from the Date: line.

$b The current date in RFC 822 format.

$c The hop count. This is a count of the number of Received: lines plus the value of the **-h** command line flag.

$d The current date in UNIX (ctime) format.

$e‡** (Obsolete; use SmtpGreetingMessage option instead.) The SMTP entry message. This is printed out when SMTP starts up. The first word must be the **$j macro as specified by RFC821. Defaults to "$j Sendmail $v ready at $b". Commonly redefined to include the configuration version number, e.g., "$j Sendmail $v/$Z ready at $b"

$f The envelope sender (from) address.

$g The sender address relative to the recipient. For example, if **$f** is "foo", **$g** will be "host!foo", "foo@host.domain", or whatever is appropriate for the receiving mailer.

$h The recipient host. This is set in ruleset 0 from the $@ field of a parsed address.

$i The queue id, e.g., "f344MXxp018717".

$j‡ The "official" domain name for this site. This is fully qualified if the full qualification can be found. It *must* be redefined to be the fully qualified domain name if your system is not configured so that information can find it automatically.

$k The UUCP node name (from the uname system call).

$l† (Obsolete; use UnixFromLine option instead.) The format of the UNIX from line. Unless you have changed the UNIX mailbox format, you should not change the default, which is "From $g $d".

$m The domain part of the *gethostname* return value. Under normal circumstances, **$j** is equivalent to **$w.$m** .

$n† The name of the daemon (for error messages). Defaults to "MAILER-DAEMON".

$o†** (Obsolete: use OperatorChars option instead.) The set of "operators" in addresses. A list of characters which will be considered tokens and which will separate tokens when doing parsing. For example, if "@" were in the **$o macro, then the input "a@b" would be scanned as three tokens: "a", "@", and "b". Defaults to ".:@[]", which is the minimum set necessary to do RFC 822 parsing; a richer set of operators is ".:%@!/[]", which adds support for UUCP, the %-hack, and X.400 addresses.

$p Sendmail's process id.

$q†** Default format of sender address. The **$q macro specifies how an address should appear in a message when it is defaulted. Defaults to "¡$g¿". It is commonly redefined to be "$?x$x ¡$g¿$—g". or "g?x ($x)$"., corresponding to the following two formats: .(b Eric Allman ¡eric@CS.Berkeley.EDU¿ eric@CS.Berkeley.EDU (Eric Allman) .)b .i Sendmail properly quotes names that have special characters if the first form is used.

$r Protocol used to receive the message. Set from the **-p** command line flag or by the SMTP server code.

$s Sender's host name. Set from the **-p** command line flag or by the SMTP server code.

$t A numeric representation of the current time.

$u The recipient user.

$v The version number of the *sendmail*binary.

$w‡ The hostname of this site. This is the root name of this host (but see below for caveats).

$x The full name of the sender.

$z The home directory of the recipient.

$_** The validated sender address. See also **${client_resolve} .

${addr_type} The type of the address which is currently being rewritten. This macro contains up to three characters, the first is either 'e' or 'h' for envelope/header address, the second is a space, and the third is either 's' or 'r' for sender/recipient address. Notice: for header addresses no distinction is currently made between sender and recipient addresses, i.e., the macro contains only 'h'.

${auth_authen} The client's authentication credentials as determined by authentication (only set if successful). The format depends on the mechanism used, it might be just 'user', or 'user@realm', or something similar (SMTP AUTH only).

${auth_author} The authorization identity, i.e. the AUTH= parameter of the .sm "SMTP MAIL" command if supplied.

${auth_type} The mechanism used for SMTP authentication (only set if successful).

${auth_ssf} The keylength (in bits) of the symmetric encryption algorithm used for the security layer of a SASL mechanism.

${bodytype} The message body type (7BIT or 8BITMIME), as determined from the envelope.

${cert_issuer} The DN (distinguished name) of the CA (certificate authority) that signed the presented certificate (the cert issuer) (STARTTLS only).

${cert_md5} The MD5 hash of the presented certificate (STARTTLS only).

${cert_subject} The DN of the presented certificate (called the cert subject) (STARTTLS only).

${cipher} The cipher suite used for the connection, e.g., EDH-DSS-DES-CBC3-SHA, EDH-RSA-DES-CBC-SHA, DES-CBC-MD5, DES-CBC3-SHA (STARTTLS only).

${cipher_bits} The keylength (in bits) of the symmetric encryption algorithm used for a TLS connection.

${client_addr} The IP address of the SMTP client. IPv6 addresses are tagged with ”IPv6:” before the address. Defined in the SMTP server only.

${client_name}** The host name of the SMTP client. This may be the client’s bracketed IP address in the form [nnn.nnn.nnn.nnn] for IPv4 and [IPv6:nnnn:...:nnnn] for IPv6 if the client’s IP address is not resolvable, or if it is resolvable but the IP address of the resolved hostname doesn’t match the original IP address. Defined in the SMTP server only. See also **${client_resolve} .

${client_port} The port number of the SMTP client. Defined in the SMTP server only.

${client_resolve}** Holds the result of the resolve call for **${client_name} . Possible values are: .(b .ta 10n OK resolved successfully FAIL permanent lookup failure FORGED forward lookup doesn’t match reverse lookup TEMP temporary lookup failure .)b Defined in the SMTP server only. .i sendmail performs a hostname lookup on the IP address of the connecting client. Next the IP addresses of that hostname are looked up. If the client IP address does not appear in that list, then the hostname is maybe forged. This is reflected as the value FORGED for **${client_resolve}** and it also shows up in **$_** as ”(may be forged)”.

${cn_issuer} The CN (common name) of the CA that signed the presented certificate (STARTTLS only).

${cn_subject} The CN (common name) of the presented certificate (STARTTLS only).

${currHeader} Header value as quoted string (possibly truncated to **MAXNAME**). This macro is only available in header check rulesets.

${daemon_addr} The IP address the daemon is listening on for connections.

${daemon_family} The network family if the daemon is accepting network connections. Possible values include “inet”, “inet6”, “iso”, “ns”, “x.25”

${daemon_flags}** The flags for the daemon as specified by the Modifier= part of **DaemonPortOptions** whereby the flags are separated from each other by spaces, and upper case flags are doubled. That is, Modifier=Ea will be represented as ”EE a” in **${daemon_flags}, which is required for testing the flags in rulesets.

${daemon_info} Some information about a daemon as a text string. For example, “SMTP+queueing@00:30:00”.

${daemon_name} The name of the daemon from **DaemonPortOptions** Name= suboption. If this suboption is not set, ”Daemon#”, where # is the daemon number, is used.

${daemon_port} The port the daemon is accepting connection on. Unless **DaemonPortOptions** is set, this will most likely be “25”.

${deliveryMode} The current delivery mode sendmail is using. It is initially set to the value of the **DeliveryMode** option.

${envid} The envelope id parameter (ENVID=) passed to sendmail as part of the envelope.

${hdrlen} The length of the header value which is stored in ${currHeader} (before possible truncation). If this value is greater than or equal to **MAXNAME** the header has been truncated.

${hdr_name} The name of the header field for which the current header check ruleset has been called. This is useful for a default header check ruleset to get the name of the header; the macro is only available in header check rulesets.

${if_addr} The IP address of the interface of an incoming connection unless it is in the loopback net. IPv6 addresses are tagged with "IPv6:" before the address.

${if_addr_out} The IP address of the interface of an outgoing connection unless it is in the loopback net. IPv6 addresses are tagged with "IPv6:" before the address.

${if_family} The IP family of the interface of an incoming connection unless it is in the loopback net.

${if_family_out} The IP family of the interface of an outgoing connection unless it is in the loopback net.

${if_name} The hostname associated with the interface of an incoming connection. This macro can be used for `SmtpGreetingMessage` and HReceived for virtual hosting. For example:

```
O SmtpGreetingMessage=$?{if_name}${if_name}$|$j$.  MTA
```

${if_name_out} The name of the interface of an outgoing connection.

${mail_addr} The address part of the resolved triple of the address given for the .sm "SMTP MAIL" command. Defined in the SMTP server only.

${mail_host} The host from the resolved triple of the address given for the .sm "SMTP MAIL" command. Defined in the SMTP server only.

${mail_mailer} The mailer from the resolved triple of the address given for the .sm "SMTP MAIL" command. Defined in the SMTP server only.

${msg_size} The value of the SIZE= parameter, i.e., usually the size of the message (in an ESMTP dialogue), before the message has been collected, thereafter the message size as computed by *sendmail* (and can be used in check_compat).

${nrcpts} The number of validated recipients for a single message. Note: since recipient validation happens after `check_rcpt` has been called, the value in this ruleset is one less than what might be expected.

${ntries} The number of delivery attempts.

${opMode} The current operation mode (from the **-b** flag).

${queue_interval}** The queue run interval given by the **-q** flag. For example, **-q30m** would set **${queue_interval} to "00:30:00".

${rcpt_addr} The address part of the resolved triple of the address given for the .sm "SMTP RCPT" command. Defined in the SMTP server only after a RCPT command.

${rcpt_host} The host from the resolved triple of the address given for the .sm "SMTP RCPT" command. Defined in the SMTP server only after a RCPT command.

${rcpt_mailer} The mailer from the resolved triple of the address given for the .sm "SMTP RCPT" command. Defined in the SMTP server only after a RCPT command.

${server_addr} The address of the server of the current outgoing SMTP connection. For LMTP delivery the macro is set to the name of the mailer.

${server_name} The name of the server of the current outgoing SMTP or LMTP connection.

${tls_version} The TLS/SSL version used for the connection, e.g., TLSv1, SSLv3, SSLv2; defined after STARTTLS has been used.

${verify} The result of the verification of the presented cert; only defined after STARTTLS has been used. Possible values are:

OK	verification succeeded.
NO	no cert presented.
NOT	no cert requested.
FAIL	cert presented but could not be verified, e.g., the signing CA is missing.
NONE	STARTTLS has not been performed.
TEMP	temporary error occurred.
PROTOCOL	some protocol error occurred.
SOFTWARE	STARTTLS handshake failed, which is a fatal error for this session, the e-mail will be queued.

14.2 All options

"AliasFile=*spec, spec, ...*" A Specify possible alias file(s). Each *spec* should be in the format "*class*: *info*" where *class*: is optional and defaults to "implicit". Note that *info* is required for all *class es* except "ldap". For the "ldap" class, if *info* is not specified, a default *info* value is used as follows:

> -k (&(objectClass=sendmailMTAAliasObject) (sendmailMTAAliasName=aliases) (—(sendmailMTACluster=$sendmailMTACluster) (sendmailMTAHost=$j)) (sendmailMTAKey=%0)) -v sendmailMTAAliasValue

Depending on how *sendmail* is compiled, valid classes are "implicit" (search through a compiled-in list of alias file types, for back compatibility), "hash" (if NEWDB is specified), "btree" (if NEWDB is specified), "dbm" (if NDBM is specified), "stab" (internal symbol table - not normally used unless you have no other database lookup), "sequence" (use a sequence of maps previously declared), "ldap" (if LDAPMAP is specified), or "nis" (if NIS is specified). If a list of *spec s* are provided, *sendmail* searches them in order.

AliasWait=*timeout* a If set, wait up to *timeout* (units default to minutes) for an "@:@" entry to exist in the alias database before starting up. If it does not appear in the *timeout* interval issue a warning.

AllowBogusHELO no short name If set, allow HELO SMTP commands that don't include a host name. Setting this violates RFC 1123 section 5.2.5, but is necessary to interoperate with several SMTP clients. If there is a value, it is still checked for legitimacy.

AuthMaxBits=*N* no short name Limit the maximum encryption strength for the security layer in SMTP AUTH (SASL). Default is essentially unlimited. This allows to turn off additional encryption in SASL if STARTTLS is already encrypting the communication, because the existing encryption strength is taken into account when choosing an algorithm for the security layer. For example, if STARTTLS is used and the symmetric cipher is 3DES, then the the keylength (in bits) is 168. Hence setting **AuthMaxBits** to 168 will disable any encryption in SASL.

AuthMechanisms no short name List of authentication mechanisms for AUTH (separated by spaces). The advertised list of authentication mechanisms will be the intersection of this list and the list of available mechanisms as determined by the Cyrus SASL library. If STARTTLS is active, EXTERNAL will be added to this list. In that case, the value of cert_subject is used as authentication id.

AuthOptions no short name List of options for SMTP AUTH consisting of single characters with intervening white space or commas.

A	Use the AUTH= parameter for the MAIL FROM command only when authentication succeeded. This can be used as a workaround for broken MTAs that do not implement RFC2554 correctly.
a	protection from active (non-dictionary) attacks during authentication exchange.
c	require mechanisms which pass client credentials, and allow mechanisms which can pass credentials to do so.
d	don't permit mechanisms susceptible to passive dictionary attack.
f	require forward secrecy between sessions (breaking one won't help break next).
p	don't permit mechanisms susceptible to simple passive attack (e.g., PLAIN, LOGIN).
y	don't permit mechanisms that allow anonymous login.

The first option applies to sendmail as a client, the others to a server. Example:

```
O AuthOptions=p,y
```

would disallow ANONYMOUS as AUTH mechanism and would allow PLAIN only if a security layer (e.g., provided by STARTTLS) is already active. The options 'a', 'c', 'd', 'f', 'p', and 'y' refer to properties of the selected SASL mechanisms. Explanations of these properties can be found in the Cyrus SASL documentation.

BadRcptThrottle=*N* no short name If set and more than the specified number of recipients in a single SMTP envelope are rejected, sleep for one second after each rejected RCPT command.

BlankSub=*c* B Set the blank substitution character to *c* . Unquoted spaces in addresses are replaced by this character. Defaults to space (i.e., no change is made).

CACERTPath no short name Path to directory with certificates of CAs. This directory directory must contain the hashes of each CA certificate as filenames (or as links to them).

CACERTFile no short name File containing one or more CA certificates; see section about STARTTLS for more information.

CheckAliases n Validate the RHS of aliases when rebuilding the alias database.

CheckpointInterval=*N* C Checkpoints the queue every *N* (default 10) addresses sent. If your system crashes during delivery to a large list, this prevents retransmission to any but the last *N* recipients.

ClassFactor=*fact* z The indicated *fact or* is multiplied by the message class (determined by the Precedence: field in the user header and the **P** lines in the configuration file) and subtracted from the priority. Thus, messages with a higher Priority: will be favored. Defaults to 1800.

ClientCertFile no short name File containing the certificate of the client, i.e., this certificate is used when *sendmail* acts as client (for STARTTLS).

ClientKeyFile no short name File containing the private key belonging to the client certificate (for STARTTLS if *sendmail* runs as client).

ClientPortOptions=*options* O Set client SMTP options. The options are *key=value* pairs separated by commas. Known keys are:

Port	Name/number of source port for connection (defaults to any free port)
Addr	Address mask (defaults INADDR_ANY)
Family	Address family (defaults to INET)
SndBufSize	Size of TCP send buffer
RcvBufSize	Size of TCP receive buffer
Modifier	Options (flags) for the daemon

The *Addr ess* mask may be a numeric address in dot notation or a network name. *Modifier* can be the following character:

h	use name of interface for HELO command
A	don't use AUTH when sending e-mail
S	don't use STARTTLS when sending e-mail

If "h" is set, the name corresponding to the outgoing interface address (whether chosen via the Connection parameter or the default) is used for the HELO/EHLO command. However, the name must not start with a square bracket and it must contain at least one dot. This is a simple test whether the name is not an IP address (in square brackets) but a qualified hostname. Note that multiple ClientPortOptions settings are allowed in order to give settings for each protocol family (e.g., one for Family=inet and one for Family=inet6). A restriction placed on one family only affects outgoing connections on that particular family.

ColonOkInAddr no short name If set, colons are acceptable in e-mail addresses (e.g., "host:user)". If not set, colons indicate the beginning of a RFC 822 group construct (¸"groupname: member1, member2, ... memberN;")". Doubled colons are always acceptable (¸"nodename::user)" and proper route-addr nesting is understood (¸"¡@relay:user@host¿)". Furthermore, this option defaults on if the configuration version level is less than 6 (for back compatibility). However, it must be off for full compatibility with RFC 822.

ConnectionCacheSize=*N* k The maximum number of open connections that will be cached at a time. The default is one. This delays closing the current connection until either this invocation of *sendmail* needs to connect to another host or it terminates. Setting it to zero defaults to the old behavior, that is, connections are closed immediately. Since this consumes file descriptors, the connection cache should be kept small: 4 is probably a practical maximum.

ConnectionCacheTimeout=*timeout* K The maximum amount of time a cached connection will be permitted to idle without activity. If this time is exceeded, the connection is immediately closed. This value should be small (on the order of ten minutes). Before *sendmail* uses a cached connection, it always sends a RSET command to check the connection; if this fails, it reopens the connection. This keeps your end from failing if the other end times out. The point of this option is to be a good network neighbor and avoid using up excessive resources on the other end. The default is five minutes.

ConnectOnlyTo=*address* no short name This can be used to override the connection address (for testing purposes).

ConnectionRateThrottle=*N* no short name If set to a positive value, allow no more than *N* incoming connections in a one second period per daemon. This is intended to flatten out peaks and allow the load average checking to cut in. Defaults to zero (no limits).

ControlSocketName=*name* no short name Name of the control socket for daemon management. A running *sendmail* daemon can be controlled through this named socket. Available commands are: *help*, *restart*, *shutdown*, and *status.* The *status* command returns the current number of daemon children, the maximum number of daemon children, the free disk space (in blocks) of the queue directory, and the load average of the machine expressed as an integer. If not set, no control socket will be available. Solaris and pre-4.4BSD kernel users should see the note in `sendmail/README`.

DHParameters File with DH parameters for STARTTLS. This is only required if a ciphersuite containing DSA/DH is used. This is only for people with a good knowledge of TLS, all others can ignore this option.

DaemonPortOptions=*options* O Set server SMTP options. Each instance of DaemonPortOptions leads to an additional incoming socket. The options are *key=value* pairs. Known keys are:

Name	User-definable name for the daemon (defaults to "Daemon#")
Port	Name/number of listening port (defaults to "smtp")
Addr	Address mask (defaults INADDR_ANY)
Family	Address family (defaults to INET)
Listen	Size of listen queue (defaults to 10)
Modifier	Options (flags) for the daemon
SndBufSize	Size of TCP send buffer
RcvBufSize	Size of TCP receive buffer

The *Name* field is used for error messages and logging. The *Address* mask may be a numeric address in dot notation or a network name. The *Family* key defaults to INET (IPv4). IPv6 users who wish to also accept IPv6 connections should add additional Family=inet6 DaemonPortOptions lines. *Modifier* can be a sequence (without any delimiters) of the following characters:

a	always require authentication
b	bind to interface through which mail has been received
c	perform hostname canonification (.cf)
f	require fully qualified hostname (.cf)
u	allow unqualified addresses (.cf)
A	disable AUTH (overrides 'a' modifier)
C	don't perform hostname canonification
E	disallow ETRN (see RFC 2476)
O	optional; if opening the socket fails ignore it
S	don't offer STARTTLS

That is, one way to specify a message submission agent (MSA) that always requires authentication is:

O DaemonPortOptions=Name=MSA, Port=587, M=Ea

The modifiers that are marked with "(.cf)" have only effect in the standard configuration file, in which they are available via **$daemon_flags .** Notice: Do **not** use the "a" modifier on a public accessible MTA! It should only be used for a MSA that is accessed by authorized users for initial mail submission. Users must authenticate to use a MSA which has this option turned on. The flags "c" and "C" can change the default for hostname canonification in the *sendmail.cf* file. See the relevant documentation for FEATURE(nocanonify) . The modifier "f" disallows addresses of the form **user@host** unless they are submitted directly. The flag "u" allows unqualified sender addresses, i.e., those without @host. "b" forces sendmail to bind to the interface through which the e-mail has been received for the outgoing connection. **WARNING:** Use "b" only if outgoing mail can be routed through the incoming connection's interface to its destination. No attempt is made to catch problems due to a misconfiguration of this parameter, use it only for virtual hosting where each virtual interface can connect to every possible location. This will also override possible settings via **ClientPortOptions.** Note, *sendmail* will listen on a new socket for each occurence of the DaemonPortOptions option in a configuration file. The modifier "O" causes sendmail to ignore a socket if it can't be opened. This applies to failures from the socket(2) and bind(2) calls.

DefaultAuthInfo no short name Filename that contains default authentication information for outgoing connections. This file must contain the user id, the authorization id, the password (plain text), the realm and the list of mechanisms to use on separate lines and must be readable by root (or the trusted user) only. If no realm is specified, `$j` is used. If no mechanisms are specified, the list given by **AuthMechanisms** is used. Notice: this option is deprecated and will be removed in future versions. Moreover, it doesn't work for the MSP since it can't read the file (the file must not be group/world-readable otherwise *sendmail* will complain). Use the authinfo ruleset instead which provides more control over the usage of the data anyway.

DefaultCharSet=*charset* no short name When a message that has 8-bit characters but is not in MIME format is converted to MIME (see the EightBitMode option) a character set must be included in the Content-Type: header. This character set is normally set from the Charset= field of the mailer descriptor. If that is not set, the value of this option is used. If this option is not set, the value "unknown-8bit" is used.

DataFileBufferSize=*threshold* no short name Set the *threshold*, in bytes, before a memory-based queue data file becomes disk-based. The default is 4096 bytes.

DeadLetterDrop=*file* no short name Defines the location of the system-wide dead.letter file, formerly hardcoded to `/usr/tmp/dead.letter`. If this option is not set (the default), sendmail will not attempt to save to a system-wide dead.letter file in the event it cannot bounce the mail to the user or postmaster. Instead, it will rename the qf file as it has in the past when the dead.letter file could not be opened.

DefaultUser=*user:group* u Set the default userid for mailers to *user:group* . If *group* is omitted and *user* is a user name (as opposed to a numeric user id) the default group listed in the `/etc/passwd` file for that user is used as the default group. Both *user* and *group* may be numeric. Mailers without the *S* flag in the mailer definition will run as this user. Defaults to 1:1. The value can also be given as a symbolic user name. [1]

DelayLA=*LA* no short name When the system load average exceeds *LA*, *sendmail* will sleep for one second on most SMTP commands and before accepting connections.

DeliverByMin=*time* 0 Set minimum time for Deliver By SMTP Service Extension (RFC 2852). If 0, no time is listed, if less than 0, the extension is not offered, if greater than 0, it is listed as minimum time for the EHLO keyword DELIVERBY.

DeliveryMode=*x* d Deliver in mode *x* . Legal modes are:

i	Deliver interactively (synchronously)
b	Deliver in background (asynchronously)
q	Just queue the message (deliver during queue run)
d	Defer delivery and all map lookups (deliver during queue run)

[1] The old **g** option has been combined into the **DefaultUser** option.

Defaults to “b” if no option is specified, “i” if it is specified but given no argument (i.e., “Od” is equivalent to “Odi”). The **-v** command line flag sets this to **i**.

DialDelay=*sleeptime* no short name Dial-on-demand network connections can see timeouts if a connection is opened before the call is set up. If this is set to an interval and a connection times out on the first connection being attempted *sendmail* will sleep for this amount of time and try again. This should give your system time to establish the connection to your service provider. Units default to seconds, so “DialDelay=5” uses a five second delay. Defaults to zero (no retry). This delay only applies to mailers which have the Z flag set.

DirectSubmissionModifiers=*modifiers* Defines **$daemon_flags** for direct (command line) submissions. If not set, **$daemon_flags** is either ”CC f” if the option **-G** is used or ”c u” otherwise.

DontBlameSendmail=*option,option,...* no short name In order to avoid possible cracking attempts caused by world- and group-writable files and directories, *sendmail* does paranoid checking when opening most of its support files. If for some reason you absolutely must run with, for example,a group-writable */etc* directory, then you will have to turn off this checking (at the cost of making your system more vulnerable to attack). The possible arguments have been described earlier. The details of these flags are described above. **”Use of this option is not recommended.”**

DontExpandCnames no short name The standards say that all host addresses used in a mail message must be fully canonical. For example, if your host is named “Cruft.Foo.ORG” and also has an alias of “FTP.Foo.ORG ,” the former name must be used at all times. This is enforced during host name canonification ($ ... $ lookups). If this option is set, the protocols are ignored and the “wrong” thing is done. However, the IETF is moving toward changing this standard, so the behavior may become acceptable. Please note that hosts downstream may still rewrite the address to be the true canonical name however.

DontInitGroups no short name If set, *sendmail* will avoid using the initgroups(3) call. If you are running NIS, this causes a sequential scan of the groups.byname map, which can cause your NIS server to be badly overloaded in a large domain. The cost of this is that the only group found for users will be their primary group (the one in the password file), which will make file access permissions somewhat more restrictive. Has no effect on systems that don’t have group lists.

DontProbeInterfaces no short name *Sendmail* normally finds the names of all interfaces active on your machine when it starts up and adds their name to the **$=w** class of known host aliases. If you have a large number of virtual interfaces or if your DNS inverse lookups are slow this can be time consuming. This option turns off that probing. However, you will need to be certain to include all variant names in the **$=w** class by some other mechanism. If set to **loopback**, loopback interfaces (e.g., lo0) will not be probed.

DontPruneRoutes R Normally, *sendmail* tries to eliminate any unnecessary explicit routes when sending an error message (as discussed in RFC 1123 section 5.2.6). For example, when sending an error message to

<@known1,@known2,@known3:user@unknown>

sendmail will strip off the “@known1,@known2” in order to make the route as direct as possible. However, if the **R** option is set, this will be disabled, and the mail will be sent to the first address in the route, even if later addresses are known. This may be useful if you are caught behind a firewall.

DoubleBounceAddress=*error-address* no short name If an error occurs when sending an error message, send the error report (termed a “double bounce” because it is an error “bounce” that occurs when trying to send another error “bounce)” to the indicated address. The address is macro expanded at the time of delivery. If not set, defaults to “postmaster”. If set to an empty string, double bounces are dropped.

EightBitMode=*action* 8 Set handling of eight-bit data. There are two kinds of eight-bit data: that declared as such using the **BODY=8BITMIME** ESMTP declaration or the **-B8BITMIME** command line flag, and undeclared 8-bit data, that is, input that just happens to be eight bits. There are three basic operations that can happen: undeclared 8-bit data can be automatically converted to 8BITMIME, undeclared 8-bit data can be passed as-is without conversion to MIME (“just send 8”), and declared 8-bit data can be converted to 7-bits for transmission to a non-8BITMIME mailer. The possible *actions* are:

s Reject undeclared 8-bit data (“strict”) do convert 8BITMIME →7BIT (“strict”)

m Convert undeclared 8-bit data to MIME (“mime”) do convert 8BITMIME → 7BIT (“mime”)

p Pass undeclared 8-bit data (“pass”)

In all cases properly declared 8BITMIME data will be converted to 7BIT as needed.

ErrorHeader=*file-or-message* E Prepend error messages with the indicated message. If it begins with a slash, it is assumed to be the pathname of a file containing a message (this is the recommended setting). Otherwise, it is a literal message. The error file might contain the name, email address, and/or phone number of a local postmaster who could provide assistance to end users. If the option is missing or null, or if it names a file which does not exist or which is not readable, no message is printed.

ErrorMode=*x* e Dispose of errors using mode *x* . The values for *x* are:

p Print error messages (default)

q No messages, just give exit status

m Mail back errors

w Write back errors (mail if user not logged in)

e Mail back errors and give zero exit stat always

FallbackMXhost=*fallbackhost* V If specified, the *fallbackhost* acts like a very low priority MX on every host. MX records will be looked up for this host, unless the name is surrounded by square brackets. This is intended to be used by sites with poor network connectivity. Messages which are undeliverable due to temporary address failures (e.g., DNS failure) also go to the FallbackMXhost.

FastSplit no short name If set to a value greater than zero (the default is one), it suppresses the MX lookups on addresses when they are initially sorted, i.e., for the first delivery attempt. This usually results in faster envelope splitting unless the MX records are readily available in a local DNS cache. To enforce initial sorting based on MX records set **FastSplit** to zero. If the mail is submitted directly from the command line, then the value also limits the number of processes to deliver the envelopes; if more envelopes are created they are only queued up and must be taken care of by a queue run. Since the default submission method is via SMTP (either from a MUA or via the MSP), the value of **FastSplit** is seldom used to limit the number of processes to deliver the envelopes.

ForkEachJob Y If set, deliver each job that is run from the queue in a separate process.

ForwardPath=*path* J Set the path for searching for users' .forward files. The default is "$z/.forward". Some sites that use the automounter may prefer to change this to "/var/forward/$u" to search a file with the same name as the user in a system directory. It can also be set to a sequence of paths separated by colons; *sendmail* stops at the first file it can successfully and safely open. For example, "/var/forward/$u:$z/.forward" will search first in /var/forward/*username* and then in *˜username /.forward* (but only if the first file does not exist).

HelpFile=*file* H Specify the help file for SMTP. If no file name is specified, "helpfile" is used.

HoldExpensive c If an outgoing mailer is marked as being expensive, don't connect immediately. This requires that queueing be compiled in, since it will depend on a queue run process to actually send the mail.

HostsFile=*path* no short name The path to the hosts database, normally "`/etc/hosts`". This option is only consulted when sendmail is canonifying addresses, and then only when "files" is in the "hosts" service switch entry. In particular, this file is *never* used when looking up host addresses; that is under the control of the system *gethostbyname (3)* routine.

HostStatusDirectory=*path* no short name The location of the long term host status information. When set, information about the status of hosts (e.g., host down or not accepting connections) will be shared between all *sendmail* processes; normally, this information is only held within a single queue run. This option requires a connection cache of at least 1 to function. If the option begins with a leading '/', it is an absolute pathname; otherwise, it is relative to the mail queue directory. A suggested value for sites desiring persistent host status is ".hoststat" (i.e., a subdirectory of the queue directory).

IgnoreDots i Ignore dots in incoming messages. This is always disabled (that is, dots are always accepted) when reading SMTP mail.

InputMailFilters=*name,name,...* A comma separated list of filters which determines which filters (see the "X - Mail Filter (Milter) Definitions" section) and the invocation sequence are contacted for incoming SMTP messages. If none are set, no filters will be contacted.

LDAPDefaultSpec=*spec* no short name Sets a default map specification for LDAP maps. The value should only contain LDAP specific settings such as "-h host -p port -d bindDN". The settings will be used for all LDAP maps unless the individual map specification overrides a setting. This option should be set before any LDAP maps are defined.

LogLevel=*n* L Set the log level to n . Defaults to 9.

M*x*—*value* no long version Set the macro x to *value* . This is intended only for use from the command line. The **-M** flag is preferred.

MailboxDatabase no short name Type of lookup to find information about local mailboxes, defaults to "pw" which uses *getpwnam.*

Other types can be introduced by adding them to the source code, see libsm/mbdb.c for details.

UseMSP no short name Use as mail submission program, i.e., allow group writable queue files if the group is the same as that of a set-group-ID sendmail binary. See the file **sendmail/SECURITY** in the distribution tarball.

MatchGECOS G Allow fuzzy matching on the GECOS field. If this flag is set, and the usual user name lookups fail (that is, there is no alias with this name and a *getpwnam* fails), sequentially search the password file for a matching entry in the GECOS field. This also requires that MATCHGECOS be turned on during compilation. This option is not recommended.

MaxAliasRecursion=*N* no short name The maximum depth of alias recursion (default: 10).

MaxDaemonChildren=*N* no short name If set, *sendmail* will refuse connections when it has more than *N* children processing incoming mail or automatic queue runs. This does not limit the number of outgoing connections. If not set, there is no limit to the number of children – that is, the system load averaging controls this.

MaxHeadersLength=*N* no short name The maximum length of the sum of all headers. This can be used to prevent a denial of service attack. The default is no limit.

MaxHopCount=*N* h The maximum hop count. Messages that have been processed more than *N* times are assumed to be in a loop and are rejected. Defaults to 25.

MaxMessageSize=*N* no short name Specify the maximum message size to be advertised in the ESMTP EHLO response. Messages larger than this will be rejected.

MaxMimeHeaderLength=*N*{/*M*} no short name Sets the maximum length of certain MIME header field values to *N* characters. These MIME header fields are determined by being a member of class checkMIMETextHeaders, which currently contains only the header Content-Description. For some of these headers which take parameters, the maximum length of each parameter is set to *M* if specified. If /*M* is not specified, one half of *N* will be used. By default, these values are 0, meaning no checks are done.

MaxQueueChildren=*N* no short name When set, this limits the number of concurrent queue runner processes to *N*. This helps to control the amount of system resources used when processing the queue. When there are multiple queue groups defined and the total number of queue runners for these queue groups would exceed *MaxQueueChildren* then the queue groups will not all run concurrently. That is, some portion of the queue groups will run concurrently such that *MaxQueueChildren* will not be exceeded, while the remaining queue groups will be run later (in round robin order). See also *MaxRunnersPerQueue* and the section **Queue Group Declaration**.

MaxQueueRunSize=*N* no short name The maximum number of jobs that will be processed in a single queue run. If not set, there is no limit on the size. If you have very large queues or a very short queue run interval this could be unstable. However, since the first *N* jobs in queue directory order are run (rather than the *N* highest priority jobs) this should be set as high as possible to avoid "losing" jobs that happen to fall late in the queue directory.

MaxRecipientsPerMessage=*N* no short name The maximum number of recipients that will be accepted per message in an SMTP transaction. Note: setting this too low can interfere with sending mail from MUAs that use SMTP for initial submission. If not set, there is no limit on the number of recipients per envelope.

MaxRunnersPerQueue=*N* no short name This sets the default maximum number of queue runners for queue groups. Up to *N* queue runners will work in parallel on a queue group's messages. This is useful where the processing of a message in the queue might delay the processing of subsequent messages. Such a delay may be the result of non-erroneous situations such as a low bandwidth connection. May be overridden on a per queue group basis by setting the *Runners* option; see the section **Queue Group Declaration**. The default is 1 when not set.

MeToo m Send to me too, even if I am in an alias expansion. This option is deprecated and will be removed from a future version.

Milter no short name This option has several sub(sub)options. The names of the suboptions are separated by dots. At the first level the following options are available:

LogLevel	Log level for input mail filter actions, defaults to LogLevel.
macros	Specifies list of macro to transmit to filters. See list below.

The “macros” option has the following suboptions which specify the list of macro to transmit to milters after a certain event occurred.

connect	After session connection start
helo	After HELO command
envfrom	After MAIL FROM command
envrcpt	After RCPT TO command

By default the lists of macros are empty. Example:

O Milter.LogLevel=12 O Milter.macros.connect=j, _, daemon_name

MinFreeBlocks=*N* b Insist on at least *N* blocks free on the filesystem that holds the queue files before accepting email via SMTP. If there is insufficient space *sendmail* gives a 452 response to the MAIL command. This invites the sender to try again later.

MinQueueAge=*age* no short name Don’t process any queued jobs that have been in the queue less than the indicated time interval. This is intended to allow you to get responsiveness by processing the queue fairly frequently without thrashing your system by trying jobs too often. The default units are minutes.

MustQuoteChars=*s* no short name Sets the list of characters that must be quoted if used in a full name that is in the phrase part of a “phrase <address>” syntax. The default is “´”.. The characters “@,;:\e()” are always added to this list.

NiceQueueRun no short name The priority of queue runners (nice(3)).

NoRecipientAction no short name The action to take when you receive a message that has no valid recipient headers (To:, Cc:, Bcc:, or Apparently-To: (the last included for back compatibility with old *sendmails*). It can be **None** to pass the message on unmodified, which violates the protocol, **Add-To** to add a To: header with any recipients it can find in the envelope (which might expose Bcc: recipients), **Add-Apparently-To** to add an Apparently-To: header (this is only for back-compatibility and is officially deprecated), **Add-To-Undisclosed** to add a header “To: undisclosed-recipients:;” to make the header legal without disclosing anything, or **Add-Bcc** to add an empty Bcc: header.

OldStyleHeaders o Assume that the headers may be in old format, i.e., spaces delimit names. This actually turns on an adaptive algorithm: if any recipient address contains a comma, parenthesis, or angle bracket, it will be assumed that commas already exist. If this flag is not on, only commas delimit names. Headers are always output with commas between the names. Defaults to off.

OperatorChars=*charlist* $o macro The list of characters that are considered to be “operators ,” that is, characters that delimit tokens. All operator characters are tokens by themselves; sequences of non-operator characters are also tokens. White space characters separate tokens but are not tokens themselves (for example, “AAA.BBB” has three tokens, but “AAA BBB” has two. If not set, OperatorChars defaults to “.—:—@—[—] ;” additionally, the characters “(—)—<—>—,—;” are always operators. Note that OperatorChars must be set in the configuration file before any rulesets.

PidFile=*filename* no short name Filename of the pid file. (default is _PATH_SENDMAILPID). The *filename* is macro-expanded before it is opened.

PostmasterCopy=*postmaster* P If set, copies of error messages will be sent to the named *postmaster* . Only the header of the failed message is sent. Errors resulting from messages with a negative precedence will not be sent. Since most errors are user problems, this is probably not a good idea on large sites, and arguably contains all sorts of privacy violations, but it seems to be popular with certain operating systems vendors. The address is macro expanded at the time of delivery. Defaults to no postmaster copies.

PrivacyOptions=—*opt,opt,...* p Set the privacy *options.* "Privacy" is really a misnomer; many of these are just a way of insisting on stricter adherence to the SMTP protocol. The *options* can be selected from:

public	Allow open access
needmailhelo	Insist on HELO or EHLO command before MAIL
needexpnhelo	Insist on HELO or EHLO command before EXPN
noexpn	Disallow EXPN entirely, implies noverb.
needvrfyhelo	Insist on HELO or EHLO command before VRFY
novrfy	Disallow VRFY entirely
noetrn	Disallow ETRN entirely
noverb	Disallow VERB entirely
restrictmailq	Restrict mailq command
restrictqrun	Restrict -q command line flag
restrictexpand	Restrict -bv and -v command line flags
noreceipts	Don't return success DSNs
nobodyreturn	Don't return the body of a message with DSNs
goaway	Disallow essentially all SMTP status queries
authwarnings	Put X-Authentication-Warning: headers in messages and log warnings

[2] The "goaway" pseudo-flag sets all flags except "noreceipts ," "restrictmailq ," "restrictqrun ," "restrictexpand ," "noetrn ," and "nobodyreturn". If mailq is restricted, only people in the same group as the queue directory can print the queue. If queue runs are restricted, only root and the owner of the queue directory can run the queue. The "restrictexpand" pseudo-flag instructs *sendmail* to drop privileges when the **-bv** option is given by users who are neither root nor the TrustedUser so users cannot read private aliases, forwards, or :include: files. It will add the "NonRootSafeAddr" to the "DontBlameSendmail" option to prevent misleading unsafe address warnings. It also overrides the **-v** (verbose) command line option to prevent information leakage. Authentication Warnings add warnings about various conditions that may indicate attempts to spoof the mail system, such as using a non-standard queue directory.

ProcessTitlePrefix=*string* no short name Prefix the process title shown on 'ps' listings with *string* . The *string* will be macro processed.

QueueDirectory=*dir* Q The QueueDirectory option serves two purposes. First, it specifies the directory or set of directories that comprise the default queue group. Second, it specifies the directory D which is the ancestor of all queue directories, and which sendmail uses as its current working directory. When sendmail dumps core, it leaves its core files in D. There are two cases. If *dir* ends with an asterisk (eg, */var/spool/mqueue/qd**), then all of the directories or symbolic links to directories beginning with 'qd' in */var/spool/mqueue* will be used as queue directories of the default queue group, and */var/spool/mqueue* will be

[2] The **noreceipts** flag turns off support for RFC 1891 (Delivery Status Notification).

used as the working directory D. Otherwise, *dir* must name a directory (usually */var/spool/mqueue*): the default queue group consists of the single queue directory *dir*, and the working directory D is set to *dir*. To define additional groups of queue directories, use the configuration file 'Q' command. Do not change the queue directory structure while sendmail is running.

QueueFactor=*factor* q Use *factor* as the multiplier in the map function to decide when to just queue up jobs rather than run them. This value is divided by the difference between the current load average and the load average limit (**QueueLA** option) to determine the maximum message priority that will be sent. Defaults to 600000.

QueueLA=*LA* x When the system load average exceeds *LA* and the **QueueFactor** (**q**) option divided by the difference in the current load average and the **QueueLA** option plus one is less than the priority of the message, just queue messages (i.e., don't try to send them). Defaults to 8 multiplied by the number of processors online on the system (if that can be determined).

QueueFileMode=*mode* no short name Default permissions for queue files (octal). If not set, sendmail uses 0600 unless its real and effective uid are different in which case it uses 0644.

QueueSortOrder=*algorithm* no short name Sets the *algorithm* used for sorting the queue. Only the first character of the value is used. Legal values are "host" (to order by the name of the first host name of the first recipient), "filename" (to order by the name of the queue file name), "time" (to order by the submission/creation time), "random" (to order randomly), "modification" (to order by the modification time of the qf file (older entries first)), and "priority" (to order by message priority). Host ordering makes better use of the connection cache, but may tend to process low priority messages that go to a single host over high priority messages that go to several hosts; it probably shouldn't be used on slow network links. Filename and modification time ordering saves the overhead of reading all of the queued items before starting the queue run. Creation (submission) time ordering is almost always a bad idea, since it allows large, bulk mail to go out before smaller, personal mail, but may have applicability on some hosts with very fast connections. Random is useful if several queue runners are started by hand which try to drain the same queue since odds are they will be working on different parts of the queue at the same time. Priority ordering is the default.

QueueTimeout=*timeout* T A synonym for "Timeout.queuereturn". Use that form instead of the "QueueTimeout" form.

RandFile no short name Name of file containing random data or the name of the UNIX socket if EGD is used. A (required) prefix "egd:" or "file:" specifies the type. STARTTLS requires this filename if the compile flag HASURANDOMDEV is not set (see sendmail/README).

ResolverOptions=*options* I Set resolver options. Values can be set using **+***flag* and cleared using **-***flag ;* the *flag s* can be "debug ," "aaonly ," "usevc ," "primary ," "igntc ," "recurse ," "defnames ," "stayopen ," "use_inet6 ," or "dnsrch". The string "HasWildcardMX" (without a **+** or **-**) can be specified to turn off matching against MX records when doing name canonifications. The string "WorkAroundBrokenAAAA" (without a **+** or **-**) can be specified to work around some broken nameservers which return SERVFAIL (a temporary failure) on T_AAAA (IPv6) lookups. Notice: it might be necessary to apply the same (or similar) options to *submit.cf* too.

RrtImpliesDsn R If this option is set, a "Return-Receipt-To:" header causes the request of a DSN, which is sent to the envelope sender as required by RFC1891, not to the address given in the header.

RunAsUser=*user* no short name The *user* parameter may be a user name (looked up in `/etc/passwd`) or a numeric user id; either form can have ":group" attached (where group can be numeric or symbolic). If set to a non-zero (non-root) value, *sendmail* will change to this user id shortly after startup[3] This avoids a certain class of security problems. However, this means that all ".forward" and ":include:" files must be readable by the indicated *user* and all files to be written must be writable by *user* Also, all file and program deliveries will be marked unsafe unless the option **DontBlameSendmail=NonRootSafeAddr** is set, in which case the delivery will be done as *user* . It is also incompatible with the **SafeFileEnvironment** option. In other words, it may not actually add much to security on an average system, and may in fact detract from security (because other file permissions must be loosened). However, it should be useful on firewalls and other places where users don't have accounts and the aliases file is well constrained.

RecipientFactor=*fact* y The indicated *fact or* is added to the priority (thus *lowering* the priority of the job) for each recipient, i.e., this value penalizes jobs with large numbers of recipients. Defaults to 30000.

RefuseLA=*LA* X When the system load average exceeds *LA*, refuse incoming SMTP connections. Defaults to 12 multiplied by the number of processors online on the system (if that can be determined).

RetryFactor=*fact* Z The *fact or* is added to the priority every time a job is processed. Thus, each time a job is processed, its priority will be decreased by the indicated value. In most environments this should be positive, since hosts that are down are all too often down for a long time. Defaults to 90000.

SafeFileEnvironment=*dir* no short name If this option is set, *sendmail* will do a *chroot (2)* call into the indicated *dir ectory* before doing any file writes. If the file name specified by the user begins with *dir*, that partial path name will be stripped off before writing, so (for example) if the SafeFileEnvironment variable is set to "/safe" then aliases of "/safe/logs/file" and "/logs/file" actually indicate the same file. Additionally, if this option is set, *sendmail* refuses to deliver to symbolic links.

SaveFromLine f Save UNIX-style "From" lines at the front of headers. Normally they are assumed redundant and discarded.

SharedMemoryKey no short name Key to use for shared memory segment; if not set (or 0), shared memory will not be used. Requires support for shared memory to be compiled into *sendmail* . If this option is set, *sendmail* can share some data between different instances. For example, the number of entries in a queue directory or the available space in a file system. This allows for more efficient program execution, since only one process needs to update the data instead of each individual process gathering the data each time it is required.

SendMimeErrors j If set, send error messages in MIME format (see RFC2045 and RFC1344 for details). If disabled, *sendmail* will not return the DSN keyword in response to an EHLO and will not do Delivery Status Notification processing as described in RFC1891.

ServerCertFile no short name File containing the certificate of the server, i.e., this certificate is used when sendmail acts as server (used for STARTTLS).

ServerKeyFile no short name File containing the private key belonging to the server certificate (used for STARTTLS).

[3] When running as a daemon, it changes to this user after accepting a connection but before reading any SMTP commands.

ServiceSwitchFile=*filename* no short name If your host operating system has a service switch abstraction (e.g., /etc/nsswitch.conf on Solaris or /etc/svc.conf on Ultrix and DEC OSF/1) that service will be consulted and this option is ignored. Otherwise, this is the name of a file that provides the list of methods used to implement particular services. The syntax is a series of lines, each of which is a sequence of words. The first word is the service name, and following words are service types. The services that *sendmail* consults directly are "aliases" and "hosts". Service types can be "dns ," "nis ," "nisplus ," or "files" (with the caveat that the appropriate support must be compiled in before the service can be referenced). If ServiceSwitchFile is not specified, it defaults to `/etc/mail/service.switch`. If that file does not exist, the default switch is:

aliases files hosts dns nis files

The default file is "`/etc/mail/service.switch`".

SevenBitInput 7 Strip input to seven bits for compatibility with old systems. This shouldn't be necessary.

SingleLineFromHeader no short name If set, From: lines that have embedded newlines are unwrapped onto one line. This is to get around a botch in Lotus Notes that apparently cannot understand legally wrapped RFC822 headers.

SingleThreadDelivery no short name If set, a client machine will never try to open two SMTP connections to a single server machine at the same time, even in different processes. That is, if another *sendmail* is already talking to some host a new *sendmail* will not open another connection. This property is of mixed value; although this reduces the load on the other machine, it can cause mail to be delayed (for example, if one *sendmail* is delivering a huge message, other *sendmail s* won't be able to send even small messages). Also, it requires another file descriptor (for the lock file) per connection, so you may have to reduce the **ConnectionCacheSize** option to avoid running out of per-process file descriptors. Requires the **HostStatusDirectory** option.

SmtpGreetingMessage=*message* $e macro The message printed when the SMTP server starts up. Defaults to "$j Sendmail $v ready at $b".

StatusFile=*file* S Log summary statistics in the named *file* . If no file name is specified, "statistics" is used. If not set, no summary statistics are saved. This file does not grow in size. It can be printed using the `mailstats` program.

SuperSafe s This option can be set to True, False, or Interactive. If set to True, *sendmail* will be super-safe when running things, i.e., always instantiate the queue file, even if you are going to attempt immediate delivery. *Sendmail* always instantiates the queue file before returning control to the client under any circumstances. This should really *always* be set to True. The Interactive value has been introduced in 8.12 and can be used together with **DeliveryMode=i .** It skips some synchronization calls which are effectively doubled in the code execution path for this mode.

TLSSrvOptions no short name List of options for SMTP STARTTLS for the server consisting of single characters with intervening white space or commas. The flag "V" disables client verification, and hence it is not possible to use a client certificate for relaying. Currently there are no other flags available.

TempFileMode=*mode* F The file mode for transcript files, files to which *sendmail* delivers directly, and files in the **HostStatusDirectory .** It is interpreted in octal by default. Defaults to 0600.

Timeout.*type*=—*timeout* r; subsumes old T option as well Set timeout values.

TimeZoneSpec=*tzinfo* t Set the local time zone info to *tzinfo* – for example, "PST8PDT". Actually, if this is not set, the TZ environment variable is cleared (so the system default is used); if set but null, the user's TZ variable is used, and if set and non-null the TZ variable is set to this value.

TrustedUser=*user* no short name The *user* parameter may be a user name (looked up in */etc/passwd)* or a numeric user id. Trusted user for file ownership and starting the daemon. If set, generated alias databases and the control socket (if configured) will automatically be owned by this user.

TryNullMXList w If this system is the "best" (that is, lowest preference) MX for a given host, its configuration rules should normally detect this situation and treat that condition specially by forwarding the mail to a UUCP feed, treating it as local, or whatever. However, in some cases (such as Internet firewalls) you may want to try to connect directly to that host as though it had no MX records at all. Setting this option causes *sendmail* to try this. The downside is that errors in your configuration are likely to be diagnosed as "host unknown" or "message timed out" instead of something more meaningful. This option is disrecommended.

UnixFromLine=*fromline* $l macro Defines the format used when *sendmail* must add a UNIX-style From_ line (that is, a line beginning "From<space>user)". Defaults to "From $g $d". Don't change this unless your system uses a different UNIX mailbox format (very unlikely).

UnsafeGroupWrites no short name If set (default), :include: and .forward files that are group writable are considered "unsafe ," that is, they cannot reference programs or write directly to files. World writable :include: and .forward files are always unsafe. Note: use **DontBlameSendmail** instead; this option is deprecated.

UseErrorsTo l If there is an "Errors-To:" header, send error messages to the addresses listed there. They normally go to the envelope sender. Use of this option causes *sendmail* to violate RFC 1123. This option is disrecommended and deprecated.

UserDatabaseSpec=*udbspec* U The user database specification.

Verbose v Run in verbose mode. If this is set, *sendmail* adjusts options **HoldExpensive** (old **c**) and **DeliveryMode** (old **d**) so that all mail is delivered completely in a single job so that you can see the entire delivery process. Option **Verbose** should *never* be set in the configuration file; it is intended for command line use only.

XscriptFileBufferSize=*threshold* no short name Set the *threshold*, in bytes, before a memory-based queue transcript file becomes disk-based. The default is 4096 bytes.

Index