

Section 48 - Log files

48. Log files

Exim writes three different logs, referred to as the main log, the reject log, and the panic log:

The main log records the arrival of each message and each delivery in a single line in each case. The format is as compact as possible, in an attempt to keep down the size of log files. Two-character flag sequences make it easy to pick out these lines. A number of other events are recorded in the main log. Some of them are optional, in which case the `log_selector` option controls whether they are included or not. A Perl script called `eximstats`, which does simple analysis of main log files, is provided in the Exim distribution (see section 49.7).

The reject log records information from messages that are rejected as a result of a configuration option (that is, for policy reasons). The first line of each rejection is a copy of the line that is also written to the main log. Then, if the message's header has been read at the time the log is written, its contents are written to this log. Only the original header lines are available; header lines added by ACLs are not logged. You can use the reject log to check that your policy controls are working correctly; on a busy host this may be easier than scanning the main log for rejection messages. You can suppress the writing of the reject log by setting `write_rejectlog` false.

When certain serious errors occur, Exim writes entries to its panic log. If the error is sufficiently disastrous, Exim bombs out afterwards. Panic log entries are usually written to the main log as well, but can get lost amid the mass of other entries. The panic log should be empty under normal circumstances. It is therefore a good idea to check it (or to have a cron script check it) regularly, in order to become aware of any problems. When Exim cannot open its panic log, it tries as a last resort to write to the system log (`syslog`). This is opened with `LOG_PID+LOG_CONS` and the facility code of `LOG_MAIL`. The message itself is written at priority `LOG_CRIT`.

Every log line starts with a timestamp, in the format shown in the following example. Note that many of the examples shown in this chapter are line-wrapped. In the log file, this would be all on one line: 2001-09-16 16:09:47 SMTP connection from [127.0.0.1] closed by QUIT

By default, the timestamps are in the local timezone. There are two ways of changing this:

You can set the `timezone` option to a different time zone; in particular, if you set `timezone = UTC`

the timestamps will be in UTC (aka GMT).

If you set `log_timezone` true, the time zone is added to the timestamp, for example: 2003-04-25 11:17:07 +0100 Start queue run: pid=12762

48.1 Where the logs are written

The logs may be written to local files, or to `syslog`, or both. However, it should be noted that many `syslog` implementations use UDP as a transport, and are therefore unreliable in the sense that messages are not guaranteed to arrive at the loghost, nor is the ordering of messages necessarily maintained. It has also been reported that on large log files (tens of megabytes) you may need to tweak `syslog` to prevent it syncing the file with each write — on Linux this has been seen to make `syslog` take 90% plus of CPU time.

The destination for Exim's logs is configured by setting `LOG_FILE_PATH` in `Local/Makefile` or by setting `log_file_path` in the run time configuration. This latter string is expanded, so it can contain, for example, references to the host name: `log_file_path = /var/log/$primary_hostname/exim_%slog`

It is generally advisable, however, to set the string in `Local/Makefile` rather than at run time, because then the setting is available right from the start of Exim's execution. Otherwise, if there's something it wants to log before it has read the configuration file (for example, an error in the configuration file) it will not use the path you want, and may not be able to log at all.

The value of LOG_FILE_PATH or log_file_path is a colon-separated list, currently limited to at most two items. This is one option where the facility for changing a list separator may not be used. The list must always be colon-separated. If an item in the list is “syslog” then syslog is used; otherwise the item must either be an absolute path, containing %s at the point where “main”, “reject”, or “panic” is to be inserted, or be empty, implying the use of a default path.

When Exim encounters an empty item in the list, it searches the list defined by LOG_FILE_PATH, and uses the first item it finds that is neither empty nor “syslog”. This means that an empty item in log_file_path can be used to mean “use the path specified at build time”. If no such item exists, log files are written in the log subdirectory of the spool directory. This is equivalent to the setting: log_file_path = \$spool_directory/log/%slog

If you do not specify anything at build time or run time, that is where the logs are written.

A log file path may also contain %D if datestamped log file names are in use – see section 48.3 below.

Here are some examples of possible settings: LOG_FILE_PATH=syslog syslog only
 LOG_FILE_PATH=:syslog syslog and default path
 LOG_FILE_PATH=syslog : /usr/log/exim_%s syslog and specified path
 LOG_FILE_PATH=/usr/log/exim_%s specified path only

If there are more than two paths in the list, the first is used and a panic error is logged. 48.2 Logging to local files that are periodically “cycled”

Some operating systems provide centralized and standardised methods for cycling log files. For those that do not, a utility script called exicyclog is provided (see section 49.6). This renames and compresses the main and reject logs each time it is called. The maximum number of old logs to keep can be set. It is suggested this script is run as a daily cron job.

An Exim delivery process opens the main log when it first needs to write to it, and it keeps the file open in case subsequent entries are required – for example, if a number of different deliveries are being done for the same message. However, remote SMTP deliveries can take a long time, and this means that the file may be kept open long after it is renamed if exicyclog or something similar is being used to rename log files on a regular basis. To ensure that a switch of log files is noticed as soon as possible, Exim calls stat() on the main log’s name before reusing an open file, and if the file does not exist, or its inode has changed, the old file is closed and Exim tries to open the main log from scratch. Thus, an old log file may remain open for quite some time, but no Exim processes should write to it once it has been renamed. 48.3 Datestamped log files

Instead of cycling the main and reject log files by renaming them periodically, some sites like to use files whose names contain a datestamp, for example, mainlog-20031225. The datestamp is in the form yyyyymmdd. Exim has support for this way of working. It is enabled by setting the log_file_path option to a path that includes %D at the point where the datestamp is required. For example: log_file_path = /var/spool/exim/log/%slog-%D
 log_file_path = /var/log/exim-%s-%D.log
 log_file_path = /var/spool/exim/log/%D-%slog

As before, %s is replaced by “main” or “reject”; the following are examples of names generated by the above examples: /var/spool/exim/log/mainlog-20021225
 /var/log/exim-reject-20021225.log
 /var/spool/exim/log/20021225-mainlog

When this form of log file is specified, Exim automatically switches to new files at midnight. It does not make any attempt to compress old logs; you will need to write your own script if you require this. You should not run exicyclog with this form of logging.

The location of the panic log is also determined by log_file_path, but it is not datestamped, because rotation of the panic log does not make sense. When generating the name of the panic log, %D is removed from the string. In addition, if it immediately follows a slash, a following non-alphanumeric character is removed; otherwise a preceding non-alphanumeric character is removed. Thus, the three examples above would give these panic log names:

/var/spool/exim/log/paniclog
 /var/log/exim-panic.log
 /var/spool/exim/log/paniclog
 48.4 Logging to syslog

The use of syslog does not change what Exim logs or the format of its messages, except in one respect. If `syslog_timestamp` is set false, the timestamps on Exim's log lines are omitted when these lines are sent to syslog. Apart from that, the same strings are written to syslog as to log files. The `syslog_facility` is set to `LOG_MAIL`, and the program name to `exim` by default, but you can change these by setting the `syslog_facility` and `syslog_processname` options, respectively. If Exim was compiled with `SYSLOG_LOG_PID` set in `Local/Makefile` (this is the default in `src/EDITME`), then, on systems that permit it (all except ULTRIX), the `LOG_PID` flag is set so that the `syslog()` call adds the pid as well as the time and host name to each line. The three log streams are mapped onto syslog priorities as follows:

-
mainlog is mapped to `LOG_INFO`

-
rejectlog is mapped to `LOG_NOTICE`

-
paniclog is mapped to `LOG_ALERT`

Many log lines are written to both mainlog and rejectlog, and some are written to both mainlog and paniclog, so there will be duplicates if these are routed by syslog to the same place. You can suppress this duplication by setting `syslog_duplication` false.

Exim's log lines can sometimes be very long, and some of its rejectlog entries contain multiple lines when headers are included. To cope with both these cases, entries written to syslog are split into separate `syslog()` calls at each internal newline, and also after a maximum of 870 data characters. (This allows for a total syslog line length of 1024, when additions such as timestamps are added.) If you are running a syslog replacement that can handle lines longer than the 1024 characters allowed by RFC 3164, you should set `SYSLOG_LONG_LINES=yes`

in `Local/Makefile` before building Exim. That stops Exim from splitting long lines, but it still splits at internal newlines in reject log entries.

To make it easy to re-assemble split lines later, each component of a split entry starts with a string of the form `[<n>/<m>]` or `[<n>\<m>]` where `<n>` is the component number and `<m>` is the total number of components in the entry. The `/` delimiter is used when the line was split because it was too long; if it was split because of an internal newline, the `\` delimiter is used. For example, supposing the length limit to be 50 instead of 870, the following would be the result of a typical rejection message to mainlog (`LOG_INFO`), each line in addition being preceded by the time, host name, and pid as added by syslog: `[1/5] 2002-09-16 16:09:43 16RdAL-0006pc-00 rejected from`
`[2/5] [127.0.0.1] (ph10): syntax error in 'From' header`
`[3/5] when scanning for sender: missing or malformed lo`
`[4/5] cal part in "<>" (envelope sender is <ph10@cam.exa`
`[5/5] mple>)`

The same error might cause the following lines to be written to `rejectlog`; (`LOG_NOTICE`): `[1/18] 2002-09-16 16:09:43 16RdAL-0006pc-00 rejected fro`
`[2/18] m [127.0.0.1] (ph10): syntax error in 'From' head`
`[3/18] er when scanning for sender: missing or malformed`
`[4/18] local part in "<>" (envelope sender is <ph10@cam`
`[5\18] .example>)`
`[6\18] Recipients: ph10@some.domain.cam.example`
`[7\18] P Received: from [127.0.0.1] (ident=ph10)`
`[8\18] by xxxxx.cam.example with smtp (Exim 4.00)`
`[9\18] id 16RdAL-0006pc-00`
`[10/18] for ph10@cam.example; Mon, 16 Sep 2002 16:`
`[11\18] 09:43 +0100`
`[12\18] F From: <>`
`[13\18] Subject: this is a test header`
`[18\18] X-something: this is another header`
`[15/18] I Message-Id: <E16RdAL-0006pc-00@xxxxx.cam.examp`
`[16\18] le>`
`[17\18] B Bcc:`
`[18/18] Date: Mon, 16 Sep 2002 16:09:43 +0100`

Log lines that are neither too long nor contain newlines are written to syslog without modification.

If only syslog is being used, the Exim monitor is unable to provide a log tail display, unless syslog is routing mainlog to a file on the local host and the environment variable EXIMON_LOG_FILE_PATH is set to tell the monitor where it is.

48.5 Log line flags

One line is written to the main log for each message received, and for each successful, unsuccessful, and delayed delivery. These lines can readily be picked out by the distinctive two-character flags that immediately follow the timestamp. The flags are:

```
<= message arrival
=> normal message delivery
-> additional address in same delivery
*> delivery suppressed by -N
** delivery failed; address bounced
== delivery deferred; temporary problem
```

48.6 Logging message reception

The format of the single-line entry in the main log that is written for every message received is shown in the basic example below, which is split over several lines in order to fit it on the page: 2002-10-31 08:57:53 16ZCW1-0005MB-00

```
<= kryten@dwarf.fict.example
H=mailer.fict.example [192.168.123.123] U=exim
P=smtp S=5678 id=<incoming message id>
```

The address immediately following “<=” is the envelope sender address. A bounce message is shown with the sender address “<”, and if it is locally generated, this is followed by an item of the form R=<message id>

which is a reference to the message that caused the bounce to be sent.

For messages from other hosts, the H and U fields identify the remote host and record the RFC 1413 identity of the user that sent the message, if one was received. The number given in square brackets is the IP address of the sending host. If there is a single, unparenthesized host name in the H field, as above, it has been verified to correspond to the IP address (see the host_lookup option). If the name is in parentheses, it was the name quoted by the remote host in the SMTP HELO or EHLO command, and has not been verified. If verification yields a different name to that given for HELO or EHLO, the verified name appears first, followed by the HELO or EHLO name in parentheses.

Misconfigured hosts (and mail forgers) sometimes put an IP address, with or without brackets, in the HELO or EHLO command, leading to entries in the log containing text like these examples: H=(10.21.32.43) [192.168.8.34]
H=([10.21.32.43]) [192.168.8.34]

This can be confusing. Only the final address in square brackets can be relied on.

For locally generated messages (that is, messages not received over TCP/IP), the H field is omitted, and the U field contains the login name of the caller of Exim.

For all messages, the P field specifies the protocol used to receive the message. This is the value that is stored in \$received_protocol. In the case of incoming SMTP messages, the value indicates whether or not any SMTP extensions (ESMTP), encryption, or authentication were used. If the SMTP session was encrypted, there is an additional X field that records the cipher suite that was used.

The protocol is set to “esmtpsa” or “esmtpa” for messages received from hosts that have authenticated themselves using the SMTP AUTH command. The first value is used when the SMTP connection was encrypted (“secure”). In this case there is an additional item A= followed by the name of the authenticator that was used. If an authenticated identification was set up by the authenticator's server_set_id option, this is logged too, separated by a colon from the authenticator name.

The id field records the existing message id, if present. The size of the received message is given by the S field. When the message is delivered, headers may be removed or added, so that the size of delivered copies of the message may not correspond with this value (and indeed may be different to each other).

The log_selector option can be used to request the logging of additional data when a message is received. See section 48.15 below. 48.7 Logging deliveries

The format of the single-line entry in the main log that is written for every delivery is shown in one of the examples below, for local and remote deliveries, respectively. Each example has been split into two lines in order to fit it on the page:

```
2002-10-31 08:59:13 16ZCW1-0005MB-00 => marv
<marv@hitch.fict.example> R=localuser T=local_delivery
2002-10-31 09:00:10 16ZCW1-0005MB-00 =>
monk@holistic.fict.example R=dnslookup T=remote_smtp
H=holistic.fict.example [192.168.234.234]
```

For ordinary local deliveries, the original address is given in angle brackets after the final delivery address, which might be a pipe or a file. If intermediate address(es) exist between the original and the final address, the last of these is given in parentheses after the final address. The R and T fields record the router and transport that were used to process the address.

If a shadow transport was run after a successful local delivery, the log line for the successful delivery has an item added on the end, of the form ST=<shadow transport name>

If the shadow transport did not succeed, the error message is put in parentheses afterwards.

When more than one address is included in a single delivery (for example, two SMTP RCPT commands in one transaction) the second and subsequent addresses are flagged with -> instead of =>. When two or more messages are delivered down a single SMTP connection, an asterisk follows the IP address in the log lines for the second and subsequent messages.

The generation of a reply message by a filter file gets logged as a “delivery” to the addressee, preceded by “>”.

The log_selector option can be used to request the logging of additional data when a message is delivered. See section 48.15 below. 48.8 Discarded deliveries

When a message is discarded as a result of the command “seen finish” being obeyed in a filter file which generates no deliveries, a log entry of the form 2002-12-10 00:50:49 16auJc-0001UB-00 => discarded <low.club@bridge.example> R=userforward

is written, to record why no deliveries are logged. When a message is discarded because it is aliased to “:blackhole:” the log line is like this: 1999-03-02 09:44:33 10HmaX-0005vi-00 => :blackhole: <hole@nowhere.example> R=blackhole_router

48.9 Deferred deliveries

When a delivery is deferred, a line of the following form is logged: 2002-12-19 16:20:23 16aiQz-0002Q5-00 == marvin@endrest.example R=dnslookup T=smtp defer (146): Connection refused

In the case of remote deliveries, the error is the one that was given for the last IP address that was tried. Details of individual SMTP failures are also written to the log, so the above line would be preceded by something like 2002-12-19 16:20:23 16aiQz-0002Q5-00 Failed to connect to mail1.endrest.example [192.168.239.239]: Connection refused

When a deferred address is skipped because its retry time has not been reached, a message is written to the log, but this can be suppressed by setting an appropriate value in log_selector. 48.10 Delivery failures

If a delivery fails because an address cannot be routed, a line of the following form is logged: 1995-12-19 16:20:23 0tRiQz-0002Q5-00 ** jim@trek99.example <jim@trek99.example>: unknown mail domain

If a delivery fails at transport time, the router and transport are shown, and the response from the remote host is included, as in this example: 2002-07-11 07:14:17 17SXDU-000189-00 ** ace400@pb.example R=dnslookup T=remote_smtp: SMTP error from remote mailer after pipelined RCPT TO:<ace400@pb.example>: host pbmail3.py.example [192.168.63.111]: 553 5.3.0

<ace400@pb.example>...Addressee unknown

The word `“pipelined”` indicates that the SMTP PIPELINING extension was being used. See `hosts_avoid_esmtp` in the `smtp` transport for a way of disabling PIPELINING. The log lines for all forms of delivery failure are flagged with `**`. 48.11 Fake deliveries

If a delivery does not actually take place because the `-N` option has been used to suppress it, a normal delivery line is written to the log, except that `“=>”` is replaced by `“*>”`. 48.12 Completion

A line of the form `2002-10-31 09:00:11 16ZCW1-0005MB-00 Completed`

is written to the main log when a message is about to be removed from the spool at the end of its processing. 48.13 Summary of Fields in Log Lines

A summary of the field identifiers that are used in log lines is shown in the following table:

A	authenticator name (and optional id)
C	SMTP confirmation on delivery
CV	certificate verification status
DN	distinguished name from peer certificate
DT	on => lines: time taken for a delivery
F	sender address (on delivery lines)
H	host name and IP address
I	local interface used
id	message id for incoming message
P	on <= lines: protocol used
	on => and ** lines: return path
QT	on => lines: time spent on queue so far
	on <code>&ldquo;Completed&rdquo;</code> lines: time spent on queue
R	on <= lines: reference for local bounce
	on => ** and == lines: router name
S	size of message
ST	shadow transport name
T	on <= lines: message subject (topic)
	on => ** and == lines: transport name
U	local user or RFC 1413 identity
X	TLS cipher suite

48.14 Other log entries

Various other types of log entry are written from time to time. Most should be self-explanatory. Among the more common are:

-

`retry time not reached` An address previously suffered a temporary error during routing or local delivery, and the time to retry has not yet arrived. This message is not written to an individual message log file unless it happens during the first delivery attempt.

-

`retry time not reached for any host` An address previously suffered temporary errors during remote delivery, and the retry time has not yet arrived for any of the hosts to which it is routed.

-

`spool file locked` An attempt to deliver a message cannot proceed because some other Exim process is already working on the message. This can be quite common if queue running processes are started at frequent intervals. The `exiwhat` utility script can be used to find out what Exim processes are doing.

-

`error ignored` There are several circumstances that give rise to this message:

-

Exim failed to deliver a bounce message whose age was greater than `ignore_bounce_errors_after`. The bounce was discarded.

-

A filter file set up a delivery using the “noerror” option, and the delivery failed. The delivery was discarded.

A delivery set up by a router configured with `errors_to = <>`

failed. The delivery was discarded. 48.15 Reducing or increasing what is logged

By setting the `log_selector` global option, you can disable some of Exim's default logging, or you can request additional logging. The value of `log_selector` is made up of names preceded by plus or minus characters. For example: `log_selector = +arguments -retry_defer`

The list of optional log items is in the following table, with the default selection marked by asterisks:

*acl_warn_skipped	skipped warn statement in ACL
address_rewrite	address rewriting
all_parents	all parents in => lines
arguments	command line arguments
*connection_reject	connection rejections
*delay_delivery	immediate delivery delayed
deliver_time	time taken to perform delivery
delivery_size	add S=nnn to => lines
*dnslist_defer	defers of DNS list (aka RBL) lookups
*etrn	ETRN commands
*host_lookup_failed	as it says
ident_timeout	timeout for ident connection
incoming_interface	incoming interface on <= lines
incoming_port	incoming port on <= lines
*lost_incoming_connection	as it says (includes timeouts)
outgoing_port	add remote port to => lines
*queue_run	start and end queue runs
queue_time	time on queue for one recipient
queue_time_overall	time on queue for whole message
received_recipients	recipients on <= lines
received_sender	sender on <= lines
*rejected_header	header contents on reject log
*retry_defer	“retry time not reached”
return_path_on_delivery	put return path on => and *\ lines
sender_on_delivery	add sender to => lines
*sender_verify_fail	sender verification failures
*size_reject	rejection because too big
*skip_delivery	delivery skipped in a queue run
smtp_confirmation	SMTP confirmation on => lines
smtp_connection	SMTP connections
smtp_incomplete_transaction	incomplete SMTP transactions
smtp_protocol_error	SMTP protocol errors
smtp_syntax_error	SMTP syntax errors
subject	contents of Subject: on <= lines
tls_certificate_verified	certificate verification status
*tls_cipher	TLS cipher suite on <= and => lines
tls_peerdn	TLS peer DN on <= and => lines
unknown_in_list	DNS lookup failed in list match
all	all of the above

More details on each of these items follows:

`acl_warn_skipped`: When an ACL warn statement is skipped because one of its conditions cannot be evaluated, a log line to this effect is written if this log selector is set.

`address_rewrite`: This applies both to global rewrites and per-transport rewrites, but not to rewrites in filters run as an unprivileged user (because such users cannot access the log).

-

`all_parents`: Normally only the original and final addresses are logged on delivery lines; with this selector, intermediate parents are given in parentheses between them.

-

`arguments`: This causes Exim to write the arguments with which it was called to the main log, preceded by the current working directory. This is a debugging feature, added to make it easier to find out how certain MUAs call `/usr/sbin/sendmail`. The logging does not happen if Exim has given up root privilege because it was called with the `-C` or `-D` options. Arguments that are empty or that contain white space are quoted. Non-printing characters are shown as escape sequences. This facility cannot log unrecognized arguments, because the arguments are checked before the configuration file is read. The only way to log such cases is to interpose a script such as `util/logargs.sh` between the caller and Exim.

-

`connection_reject`: A log entry is written whenever an incoming SMTP connection is rejected, for whatever reason.

-

`delay_delivery`: A log entry is written whenever a delivery process is not started for an incoming message because the load is too high or too many messages were received on one connection. Logging does not occur if no delivery process is started because `queue_only` is set or `-odq` was used.

-

`deliver_time`: For each delivery, the amount of real time it has taken to perform the actual delivery is logged as `DT=<time>`, for example, `DT=1s`.

-

`delivery_size`: For each delivery, the size of message delivered is added to the `“=>”` line, tagged with `S=`.

-

`dnslist_defer`: A log entry is written if an attempt to look up a host in a DNS black list suffers a temporary error.

-

`etrn`: Every legal ETRN command that is received is logged, before the ACL is run to determine whether or not it is actually accepted. An invalid ETRN command, or one received within a message transaction is not logged by this selector (see `smtp_syntax_error` and `smtp_protocol_error`).

-

`host_lookup_failed`: When a lookup of a host's IP addresses fails to find any addresses, or when a lookup of an IP address fails to find a host name, a log line is written. This logging does not apply to direct DNS lookups when routing email addresses, but it does apply to `“byname”` lookups.

-

`ident_timeout`: A log line is written whenever an attempt to connect to a client's ident port times out.

-

`incoming_interface`: The interface on which a message was received is added to the `“<=”` line as an IP address in square brackets, tagged by `I=` and followed by a colon and the port number. The local interface and port are also added to other SMTP log lines, for example `“SMTP connection from”`, and to rejection lines.

-

`incoming_port`: The remote port number from which a message was received is added to log entries and `Received:` header lines, following the IP address in square brackets, and separated from it by a colon. This is implemented by changing the value that is put in the `$sender_fullhost` and `$sender_rcvhost` variables. Recording the remote port number has become more important with the widening use of NAT (see RFC 2505).

-

`lost_incoming_connection`: A log line is written when an incoming SMTP connection is unexpectedly dropped.

-

`outgoing_port`: The remote port number is added to delivery log lines (those containing `=>` tags) following the IP address. This option is not included in the default setting, because for most ordinary configurations, the remote port number is always 25 (the SMTP port).

-

queue_run: The start and end of every queue run are logged.

queue_time: The amount of time the message has been in the queue on the local host is logged as QT=<time> on delivery (=) lines, for example, QT=3m45s. The clock starts when Exim starts to receive the message, so it includes reception time as well as the delivery time for the current address. This means that it may be longer than the difference between the arrival and delivery log line times, because the arrival log line is not written until the message has been successfully received.

queue_time_overall: The amount of time the message has been in the queue on the local host is logged as QT=<time> on "Completed" lines, for example, QT=3m45s. The clock starts when Exim starts to receive the message, so it includes reception time as well as the total delivery time.

received_recipients: The recipients of a message are listed in the main log as soon as the message is received. The list appears at the end of the log line that is written when a message is received, preceded by the word "for:". The addresses are listed after they have been qualified, but before any rewriting has taken place. Recipients that were discarded by an ACL for MAIL or RCPT do not appear in the list.

received_sender: The unrewritten original sender of a message is added to the end of the log line that records the message's arrival, after the word "from:" (before the recipients if received_recipients is also set).

rejected_header: If a message's header has been received at the time a rejection is written to the reject log, the complete header is added to the log. Header logging can be turned off individually for messages that are rejected by the local_scan() function (see section 41.2).

retry_defer: A log line is written if a delivery is deferred because a retry time has not yet been reached. However, this "retry time not reached" message is always omitted from individual message logs after the first delivery attempt.

return_path_on_delivery: The return path that is being transmitted with the message is included in delivery and bounce lines, using the tag P=. This is omitted if no delivery actually happens, for example, if routing fails, or if delivery is to /dev/null or to :blackhole:.

sender_on_delivery: The message's sender address is added to every delivery and bounce line, tagged by F= (for "from:"). This is the original sender that was received with the message; it is not necessarily the same as the outgoing return path.

sender_verify_failure: If this selector is unset, the separate log line that gives details of a sender verification failure is not written. Log lines for the rejection of SMTP commands contain just "sender verify failed", so some detail is lost.

size_reject: A log line is written whenever a message is rejected because it is too big.

skip_delivery: A log line is written whenever a message is skipped during a queue run because it is frozen or because another process is already delivering it. The message that is written is "spool file is locked".

smtp_confirmation: The response to the final " " in the SMTP dialogue for outgoing messages is added to delivery log lines in the form C=<text>. A number of MTAs (including Exim) return an identifying string in this response.

smtp_connection: A log line is written whenever an SMTP connection is established or closed, unless the connection is from a host that matches hosts_connection_nolog. (In contrast, lost_incoming_connection applies only when the closure is unexpected.) This applies to connections from local processes that use -bs as well as to TCP/IP connections. If a connection is dropped in the middle of a message, a log line is always written, whether or not this selector is set, but

otherwise nothing is written at the start and end of connections unless this selector is enabled.

For TCP/IP connections to an Exim daemon, the current number of connections is included in the log message for each new connection, but note that the count is reset if the daemon is restarted. Also, because connections are closed (and the closure is logged) in subprocesses, the count may not include connections that have been closed but whose termination the daemon has not yet noticed. Thus, while it is possible to match up the opening and closing of connections in the log, the value of the logged counts may not be entirely accurate.

`smtp_incomplete_transaction`: When a mail transaction is aborted by RSET, QUIT, loss of connection, or otherwise, the incident is logged, and the message sender plus any accepted recipients are included in the log line. This can provide evidence of dictionary attacks.

`smtp_protocol_error`: A log line is written for every SMTP protocol error encountered. Exim does not have perfect detection of all protocol errors because of transmission delays and the use of pipelining. If PIPELINING has been advertised to a client, an Exim server assumes that the client will use it, and therefore it does not count "unexpected" errors (for example, RCPT received after rejecting MAIL) as protocol errors.

`smtp_syntax_error`: A log line is written for every SMTP syntax error encountered. An unrecognized command is treated as a syntax error. For an external connection, the host identity is given; for an internal connection using `-bs` the sender identification (normally the calling user) is given.

`subject`: The subject of the message is added to the arrival log line, preceded by "T=" (T for "topic", since S is already used for "size"). Any MIME "words" in the subject are decoded. The `print_topbitchars` option specifies whether characters with values greater than 127 should be logged unchanged, or whether they should be rendered as escape sequences.

`tls_certificate_verified`: An extra item is added to `<=` and `=>` log lines when TLS is in use. The item is `CV=yes` if the peer's certificate was verified, and `CV=no` if not.

`tls_cipher`: When a message is sent or received over an encrypted connection, the cipher suite used is added to the log line, preceded by `X=`.

`tls_peerdn`: When a message is sent or received over an encrypted connection, and a certificate is supplied by the remote host, the peer DN is added to the log line, preceded by `DN=`.

`unknown_in_list`: This setting causes a log entry to be written when the result of a list match is failure because a DNS lookup failed. 48.16 Message log

In addition to the general log files, Exim writes a log file for each message that it handles. The names of these per-message logs are the message ids, and they are kept in the `msglog` sub-directory of the spool directory. Each message log contains copies of the log lines that apply to the message. This makes it easier to inspect the status of an individual message without having to search the main log. A message log is deleted when processing of the message is complete, unless `preserve_message_logs` is set, but this should be used only with great care because they can fill up your disk very quickly. On a heavily loaded system, it may be desirable to disable the use of per-message logs, in order to reduce disk I/O. This can be done by setting the `message_logs` option false.