

## Section 44 - SMTP processing

### 44. SMTP processing

Exim supports a number of different ways of using the SMTP protocol, and its LMTP variant, which is an interactive protocol for transferring messages into a closed mail store application. This chapter contains details of how SMTP is processed. For incoming mail, the following are available:

-

SMTP over TCP/IP (Exim daemon or inetd);

-

SMTP over the standard input and output (the `-bs` option);

-

Batched SMTP on the standard input (the `-bS` option).

For mail delivery, the following are available:

-

SMTP over TCP/IP (the `smtp` transport);

-

LMTP over TCP/IP (the `smtp` transport with the `protocol` option set to `"lmtp"`);

-

LMTP over a pipe to a process running in the local host (the `lmtp` transport);

-

Batched SMTP to a file or pipe (the `appendfile` and `pipe` transports with the `use_bsmtpl` option set).

Batched SMTP is the name for a process in which batches of messages are stored in or read from files (or pipes), in a format in which SMTP commands are used to contain the envelope information. 44.1 Outgoing SMTP and LMTP over TCP/IP

Outgoing SMTP and LMTP over TCP/IP is implemented by the `smtp` transport. The `protocol` option selects which protocol is to be used, but the actual processing is the same in both cases.

If, in response to its EHLO command, Exim is told that the `SIZE` parameter is supported, it adds `SIZE=<n>` to each subsequent MAIL command. The value of `<n>` is the message size plus the value of the `size_addition` option (default 1024) to allow for additions to the message such as per-transport header lines, or changes made in a transport filter. If `size_addition` is set negative, the use of `SIZE` is suppressed.

If the remote server advertises support for PIPELINING, Exim uses the pipelining extension to SMTP (RFC 2197) to reduce the number of TCP/IP packets required for the transaction.

If the remote server advertises support for the STARTTLS command, and Exim was built to support TLS encryption, it tries to start a TLS session unless the server matches `hosts_avoid_tls`. See chapter 38 for more details.

If the remote server advertises support for the AUTH command, Exim scans the authenticators configuration for any suitable client settings, as described in chapter 33.

Responses from the remote host are supposed to be terminated by CR followed by LF. However, there are known to be hosts that do not send CR characters, so in order to be able to interwork with such hosts, Exim treats LF on its own as a line terminator.

If a message contains a number of different addresses, all those with the same characteristics (for example, the same envelope sender) that resolve to the same set of hosts, in the same order, are sent in a single SMTP transaction, even if they are for different domains, unless there are more than the setting of the `max_rcpts` option in the `smtp` transport allows, in which case they are split into groups containing no more than `max_rcpts` addresses each. If `remote_max_parallel` is greater than one, such groups may be sent in parallel sessions. The order of hosts with identical MX values is not significant when checking whether addresses can be batched in this way.

When the `smtp` transport suffers a temporary failure that is not message-related, Exim updates its transport-specific

database, which contains records indexed by host name that remember which messages are waiting for each particular host. It also updates the retry database with new retry times.

Exim's retry hints are based on host name plus IP address, so if one address of a multi-homed host is broken, it will soon be skipped most of the time. See the next section for more detail about error handling.

When a message is successfully delivered over a TCP/IP SMTP connection, Exim looks in the hints database for the transport to see if there are any queued messages waiting for the host to which it is connected. If it finds one, it creates a new Exim process using the -MC option (which can only be used by a process running as root or the Exim user) and passes the TCP/IP socket to it so that it can deliver another message using the same socket. The new process does only those deliveries that are routed to the connected host, and may in turn pass the socket on to a third process, and so on.

The connection\_max\_messages option of the smtp transport can be used to limit the number of messages sent down a single TCP/IP connection.

The second and subsequent messages delivered down an existing connection are identified in the main log by the addition of an asterisk after the closing square bracket of the IP address. 44.2 Errors in outgoing SMTP

Three different kinds of error are recognized for outgoing SMTP: host errors, message errors, and recipient errors. Host errors

A host error is not associated with a particular message or with a particular recipient of a message. The host errors are:

-

Connection refused or timed out,

-

Any error response code on connection,

-

Any error response code to EHLO or HELO,

-

Loss of connection at any time, except after ":",

-

I/O errors at any time,

-

Timeouts during the session, other than in response to MAIL, RCPT or the ":" at the end of the data.

For a host error, a permanent error response on connection, or in response to EHLO, causes all addresses routed to the host to be failed. Any other host error causes all addresses to be deferred, and retry data to be created for the host. It is not tried again, for any message, until its retry time arrives. If the current set of addresses are not all delivered in this run (to some alternative host), the message is added to the list of those waiting for this host, so if it is still undelivered when a subsequent successful delivery is made to the host, it will be sent down the same SMTP connection. Message errors

A message error is associated with a particular message when sent to a particular host, but not with a particular recipient of the message. The message errors are:

-

Any error response code to MAIL, DATA, or the ":" that terminates the data,

-

Timeout after MAIL,

-

Timeout or loss of connection after the ":" that terminates the data. A timeout after the DATA command itself is treated as a host error, as is loss of connection at any other time.

For a message error, a permanent error response (5xx) causes all addresses to be failed, and a delivery error report to be returned to the sender. A temporary error response (4xx), or one of the timeouts, causes all addresses to be deferred. Retry data is not created for the host, but instead, a retry record for the combination of host plus message id is created. The message is not added to the list of those waiting for this host. This ensures that the failing message will not be sent to this host again until the retry time arrives. However, other messages that are routed to the host are not affected, so if it

is some property of the message that is causing the error, it will not stop the delivery of other mail.

If the remote host specified support for the SIZE parameter in its response to EHLO, Exim adds SIZE=nnn to the MAIL command, so an over-large message will cause a message error because the error arrives as a response to MAIL.

#### Recipient errors

A recipient error is associated with a particular recipient of a message. The recipient errors are:

-

Any error response to RCPT,

-

Timeout after RCPT.

For a recipient error, a permanent error response (5xx) causes the recipient address to be failed, and a bounce message to be returned to the sender. A temporary error response (4xx) or a timeout causes the failing address to be deferred, and routing retry data to be created for it. This is used to delay processing of the address in subsequent queue runs, until its routing retry time arrives. This applies to all messages, but because it operates only in queue runs, one attempt will be made to deliver a new message to the failing address before the delay starts to operate. This ensures that, if the failure is really related to the message rather than the recipient (‘message too big for this recipient’ is a possible example), other messages have a chance of getting delivered. If a delivery to the address does succeed, the retry information gets cleared, so all stuck messages get tried again, and the retry clock is reset.

The message is not added to the list of those waiting for this host. Use of the host for other messages is unaffected, and except in the case of a timeout, other recipients are processed independently, and may be successfully delivered in the current SMTP session. After a timeout it is of course impossible to proceed with the session, so all addresses get deferred. However, those other than the one that failed do not suffer any subsequent retry delays. Therefore, if one recipient is causing trouble, the others have a chance of getting through when a subsequent delivery attempt occurs before the failing recipient’s retry time.

In all cases, if there are other hosts (or IP addresses) available for the current set of addresses (for example, from multiple MX records), they are tried in this run for any undelivered addresses, subject of course to their own retry data. In other words, recipient error retry data does not take effect until the next delivery attempt.

Some hosts have been observed to give temporary error responses to every MAIL command at certain times (‘insufficient space’ has been seen). It would be nice if such circumstances could be recognized, and defer data for the host itself created, but this is not possible within the current Exim design. What actually happens is that retry data for every (host, message) combination is created.

The reason that timeouts after MAIL and RCPT are treated specially is that these can sometimes arise as a result of the remote host’s verification procedures. Exim makes this assumption, and treats them as if a temporary error response had been received. A timeout after ‘.’ is treated specially because it is known that some broken implementations fail to recognize the end of the message if the last character of the last line is a binary zero. Thus, it is helpful to treat this case as a message error.

Timeouts at other times are treated as host errors, assuming a problem with the host, or the connection to it. If a timeout after MAIL, RCPT, or ‘.’ is really a connection problem, the assumption is that at the next try the timeout is likely to occur at some other point in the dialogue, causing it then to be treated as a host error.

There is experimental evidence that some MTAs drop the connection after the terminating ‘.’ if they do not like the contents of the message for some reason, in contravention of the RFC, which indicates that a 5xx response should be given. That is why Exim treats this case as a message rather than a host error, in order not to delay other messages to the same host.

#### 44.3 Incoming SMTP messages over TCP/IP

Incoming SMTP messages can be accepted in one of two ways: by running a listening daemon, or by using inetd. In the latter case, the entry in /etc/inetd.conf should be like this: smtp stream tcp nowait exim /opt/exim/bin/exim in.exim -bs

Exim distinguishes between this case and the case of a locally running user agent using the -bs option by checking whether or not the standard input is a socket. When it is, either the port must be privileged (less than 1024), or the caller must be root or the Exim user. If any other user passes a socket with an unprivileged port number, Exim prints a message on the standard error stream and exits with an error code.

By default, Exim does not make a log entry when a remote host connects or disconnects (either via the daemon or inetd), unless the disconnection is unexpected. It can be made to write such log entries by setting the smtp\_connection

log selector.

Commands from the remote host are supposed to be terminated by CR followed by LF. However, there are known to be hosts that do not send CR characters. In order to be able to interwork with such hosts, Exim treats LF on its own as a line terminator. Furthermore, because common code is used for receiving messages from all sources, a CR on its own is also interpreted as a line terminator. However, the sequence &ldquo;CR, dot, CR&rdquo; does not terminate incoming SMTP data.

One area that sometimes gives rise to problems concerns the EHLO or HELO commands. Some clients send syntactically invalid versions of these commands, which Exim rejects by default. (This is nothing to do with verifying the data that is sent, so `helo_verify_hosts` is not relevant.) You can tell Exim not to apply a syntax check by setting `helo_accept_junk_hosts` to match the broken hosts that send invalid commands.

The amount of disk space available is checked whenever SIZE is received on a MAIL command, independently of whether `message_size_limit` or `check_spool_space` is configured, unless `smtp_check_spool_space` is set false. A temporary error is given if there is not enough space. If `check_spool_space` is set, the check is for that amount of space plus the value given with SIZE, that is, it checks that the addition of the incoming message will not reduce the space below the threshold.

When a message is successfully received, Exim includes the local message id in its response to the final &ldquo;&rdquo; that terminates the data. If the remote host logs this text it can help with tracing what has happened to a message.

The Exim daemon can limit the number of simultaneous incoming connections it is prepared to handle (see the `smtp_accept_max` option). It can also limit the number of simultaneous incoming connections from a single remote host (see the `smtp_accept_max_per_host` option). Additional connection attempts are rejected using the SMTP temporary error code 421.

The Exim daemon does not rely on the SIGCHLD signal to detect when a subprocess has finished, as this can get lost at busy times. Instead, it looks for completed subprocesses every time it wakes up. Provided there are other things happening (new incoming calls, starts of queue runs), completed processes will be noticed and tidied away. On very quiet systems you may sometimes see a &ldquo;defunct&rdquo; Exim process hanging about. This is not a problem; it will be noticed when the daemon next wakes up.

When running as a daemon, Exim can reserve some SMTP slots for specific hosts, and can also be set up to reject SMTP calls from non-reserved hosts at times of high system load &ndash; for details see the `smtp_accept_reserve`, `smtp_load_reserve`, and `smtp_reserve_hosts` options. The load check applies in both the daemon and `inetd` cases.

Exim normally starts a delivery process for each message received, though this can be varied by means of the `-odq` command line option and the `queue_only`, `queue_only_file`, and `queue_only_load` options. The number of simultaneously running delivery processes started in this way from SMTP input can be limited by the `smtp_accept_queue` and `smtp_accept_queue_per_connection` options. When either limit is reached, subsequently received messages are just put on the input queue without starting a delivery process.

The controls that involve counts of incoming SMTP calls (`smtp_accept_max`, `smtp_accept_queue`, `smtp_accept_reserve`) are not available when Exim is started up from the `inetd` daemon, because in that case each connection is handled by an entirely independent Exim process. Control by load average is, however, available with `inetd`.

Exim can be configured to verify addresses in incoming SMTP commands as they are received. See chapter 39 for details. It can also be configured to rewrite addresses at this time &ndash; before any syntax checking is done. See section 31.9.

Exim can also be configured to limit the rate at which a client host submits MAIL and RCPT commands in a single SMTP session. See the `smtp_ratelimit_hosts` option. 44.4 Unrecognized SMTP commands

If Exim receives more than `smtp_max_unknown_commands` unrecognized SMTP commands during a single SMTP connection, it drops the connection after sending the error response to the last command. The default value for `smtp_max_unknown_commands` is 3. This is a defence against some kinds of abuse that subvert web servers into making connections to SMTP ports; in these circumstances, a number of non-SMTP lines are sent first. 44.5 Syntax and protocol errors in SMTP commands

A syntax error is detected if an SMTP command is recognized, but there is something syntactically wrong with its data, for example, a malformed email address in a RCPT command. Protocol errors include invalid command sequencing such as RCPT before MAIL. If Exim receives more than `smtp_max_synprot_errors` such commands during a single SMTP

connection, it drops the connection after sending the error response to the last command. The default value for `smtp_max_synprot_errors` is 3. This is a defence against broken clients that loop sending bad commands (yes, it has been seen).

#### 44.6 Use of non-mail SMTP commands

The “non-mail” SMTP commands are those other than MAIL, RCPT, and DATA. Exim counts such commands, and drops the connection if there are too many of them in a single SMTP session. This action catches some denial-of-service attempts and things like repeated failing AUTHs, or a mad client looping sending EHLO. The global option `smtp_accept_max_nonmail` defines what “too many” means. Its default value is 10.

When a new message is expected, one occurrence of RSET is not counted. This allows a client to send one RSET between messages (this is not necessary, but some clients do it). Exim also allows one uncounted occurrence of HELO or EHLO, and one occurrence of STARTTLS between messages. After starting up a TLS session, another EHLO is expected, and so it too is not counted.

The first occurrence of AUTH in a connection, or immediately following STARTTLS is also not counted. Otherwise, all commands other than MAIL, RCPT, DATA, and QUIT are counted.

You can control which hosts are subject to the limit set by `smtp_accept_max_nonmail` by setting `smtp_accept_max_nonmail_hosts`. The default value is \*, which makes the limit apply to all hosts. This option means that you can exclude any specific badly-behaved hosts that you have to live with.

#### 44.7 The VRFY and EXPN commands

When Exim receives a VRFY or EXPN command on a TCP/IP connection, it runs the ACL specified by `acl_smtp_vrfy` or `acl_smtp_expn` (as appropriate) in order to decide whether the command should be accepted or not. If no ACL is defined, the command is rejected.

When VRFY is accepted, it runs exactly the same code as when Exim is called with the `-bv` option.

When EXPN is accepted, a single-level expansion of the address is done. EXPN is treated as an “address test” (similar to the `-bt` option) rather than a verification (the `-bv` option). If an unqualified local part is given as the argument to EXPN, it is qualified with `qualify_domain`. Rejections of VRFY and EXPN commands are logged on the main and reject logs, and VRFY verification failures are logged on the main log for consistency with RCPT failures.

#### 44.8 The ETRN command

RFC 1985 describes an SMTP command called ETRN that is designed to overcome the security problems of the TURN command (which has fallen into disuse). When Exim receives an ETRN command on a TCP/IP connection, it runs the ACL specified by `acl_smtp_etrn` in order to decide whether the command should be accepted or not. If no ACL is defined, the command is rejected.

The ETRN command is concerned with “releasing” messages that are awaiting delivery to certain hosts. As Exim does not organize its message queue by host, the only form of ETRN that is supported by default is the one where the text starts with the “#” prefix, in which case the remainder of the text is specific to the SMTP server. A valid ETRN command causes a run of Exim with the `-R` option to happen, with the remainder of the ETRN text as its argument. For example, `ETRN #brigadoon`

runs the command `exim -R brigadoon`

which causes a delivery attempt on all messages with undelivered addresses containing the text “brigadoon”. When `smtp_etrn_serialize` is set (the default), Exim prevents the simultaneous execution of more than one queue run for the same argument string as a result of an ETRN command. This stops a misbehaving client from starting more than one queue runner at once.

Exim implements the serialization by means of a hints database in which a record is written whenever a process is started by ETRN, and deleted when the process completes. However, Exim does not keep the SMTP session waiting for the ETRN process to complete. Once ETRN is accepted, the client is sent a “success” return code. Obviously there is scope for hints records to get left lying around if there is a system or program crash. To guard against this, Exim ignores any records that are more than six hours old.

For more control over what ETRN does, the `smtp_etrn_command` option can be used. This specifies a command that is run whenever ETRN is received, whatever the form of its argument. For example: `smtp_etrn_command = /etc/etrn_command $domain \ $sender_host_address`

The string is split up into arguments which are independently expanded. The expansion variable `$domain` is set to the argument of the ETRN command, and no syntax checking is done on the contents of this argument. Exim does not wait for the command to complete, so its status code is not checked. Exim runs under its own uid and gid when receiving incoming SMTP, so it is not possible for it to change them before running the command. 44.9 Incoming local SMTP

Some user agents use SMTP to pass messages to their local MTA using the standard input and output, as opposed to passing the envelope on the command line and writing the message to the standard input. This is supported by the `-bs` option. This form of SMTP is handled in the same way as incoming messages over TCP/IP (including the use of ACLs), except that the envelope sender given in a MAIL command is ignored unless the caller is trusted. In an ACL you can detect this form of SMTP input by testing for an empty host identification. It is common to have this as the first line in the ACL that runs for RCPT commands: `accept hosts = :`

This accepts SMTP messages from local processes without doing any other tests. 44.10 Outgoing batched SMTP

Both the appendfile and pipe transports can be used for handling batched SMTP. Each has an option called `use_bsmtmp` which causes messages to be output in BSMTP format. No SMTP responses are possible for this form of delivery. All it is doing is using SMTP commands as a way of transmitting the envelope along with the message.

The message is written to the file or pipe preceded by the SMTP commands MAIL and RCPT, and followed by a line containing a single dot. Lines in the message that start with a dot have an extra dot added. The SMTP command HELO is not normally used. If it is required, the `message_prefix` option can be used to specify it.

Because appendfile and pipe are both local transports, they accept only one recipient address at a time by default. However, you can arrange for them to handle several addresses at once by setting the `batch_max` option. When this is done for BSMTP, messages may contain multiple RCPT commands. See chapter 25 for more details.

When one or more addresses are routed to a BSMTP transport by a router that sets up a host list, the name of the first host on the list is available to the transport in the variable `$host`. Here is an example of such a transport and router:

```
begin routers
route_append:
  driver = manualroute
  transport = smtp_appendfile
  route_list = domain.example batch.host.example

begin transports
smtp_appendfile:
  driver = appendfile
  directory = /var/bsmtmp/$host
  batch_max = 1000
  use_bsmtmp
  user = exim
```

This causes messages addressed to `domain.example` to be written in BSMTP format to `/var/bsmtmp/batch.host.example`, with only a single copy of each message (unless there are more than 1000 recipients). 44.11 Incoming batched SMTP

The `-bS` command line option causes Exim to accept one or more messages by reading SMTP on the standard input, but to generate no responses. If the caller is trusted, the senders in the MAIL commands are believed; otherwise the sender is always the caller of Exim. Unqualified senders and receivers are not rejected (there seems little point) but instead just get qualified. HELO and EHLO act as RSET; VRFY, EXPN, ETRN and HELP, act as NOOP; QUIT quits.

No policy checking is done for BSMTP input. That is, no ACL is run at anytime. In this respect it is like non-SMTP local input.

If an error is detected while reading a message, including a missing `&ldquo;`; `&rdquo;`; at the end, Exim gives up immediately. It writes details of the error to the standard output in a stylized way that the calling program should be able to make some use of automatically, for example: 554 Unexpected end of file

```
Transaction started in line 10
Error detected in line 14
```

It writes a more verbose version, for human consumption, to the standard error file, for example: An error was detected while processing a file of BSMTP input.

The error message was:

501 '>' missing at end of address

The SMTP transaction started in line 10.

The error was detected in line 12.

The SMTP command at fault was:

rcpt to:<malformed@in.com.plete

1 previous message was successfully processed.

The rest of the batch was abandoned.

The return code from Exim is zero only if there were no errors. It is 1 if some messages were accepted before an error was detected, and 2 if no messages were accepted.