

## Section 52 - Format of spool files

### 52. Format of spool files

A message on Exim's queue consists of two files, whose names are the message id followed by -D and -H, respectively. The data portion of the message is kept in the -D file on its own. The message's envelope, status, and headers are all kept in the -H file, whose format is described in this chapter. Each of these two files contains the final component of its own name as its first line. This is insurance against disk crashes where the directory is lost but the files themselves are recoverable.

Some people are tempted into editing -D files in order to modify messages. You need to be extremely careful if you do this; it is not recommended and you are on your own if you do it. Here are some of the pitfalls:

-

You must ensure that Exim does not try to deliver the message while you are fiddling with it. The safest way is to take out a write lock on the -D file, which is what Exim itself does, using `fcntl()`. If you update the file in place, the lock will be retained. If you write a new file and rename it, the lock will be lost at the instant of rename.

-

If you change the number of lines in the file, the value of `$body_linecount`, which is stored in the -H file, will be incorrect. At present, this value is not used by Exim, but there is no guarantee that this will always be the case.

-

If the message is in MIME format, you must take care not to break it.

-

If the message is cryptographically signed, any change will invalidate the signature.

Files whose names end with -J may also be seen in the input directory (or its subdirectories when `split_spool_directory` is set). These are journal files, used to record addresses to which the message has been delivered during the course of a delivery run. At the end of the run, the -H file is updated, and the -J file is deleted. 52.1 Format of the -H file

The second line of the -H file contains the login name for the uid of the process that called Exim to read the message, followed by the numerical uid and gid. For a locally generated message, this is normally the user who sent the message. For a message received over TCP/IP, it is normally the Exim user.

The third line of the file contains the address of the message's sender as transmitted in the envelope, contained in angle brackets. The sender address is empty for bounce messages. For incoming SMTP mail, the sender address is given in the MAIL command. For locally generated mail, the sender address is created by Exim from the login name of the current user and the configured `qualify_domain`. However, this can be overridden by the -f option or a leading `&ldquo;From &rdquo;` line if the caller is trusted, or if the supplied address is `&ldquo;<>&rdquo;`; or an address that matches `untrusted_set_senders`.

The fourth line contains two numbers. The first is the time that the message was received, in the conventional Unix form `&ndash;`; the number of seconds since the start of the epoch. The second number is a count of the number of messages warning of delayed delivery that have been sent to the sender.

There follow a number of lines starting with a hyphen. These can appear in any order, and are omitted when not relevant: `-acl <number> <length>`

This item is obsolete, and is not generated from Exim release 4.61 onwards; `-aclc` and `-aclm` are used instead. However, `-acl` is still recognized, to provide backward compatibility. In the old format, a line of this form is present for every ACL variable that is not empty. The number identifies the variable; the `acl_cx` variables are numbered 0&ndash;9 and the `acl_mx` variables are numbered 10&ndash;19. The length is the length of the data string for the variable. The string itself starts at the beginning of the next line, and is followed by a newline character. It may contain internal newlines. `-aclc <number> <length>`

A line of this form is present for every ACL connection variable that is not empty. The number identifies the variable. The length is the length of the data string for the variable. The string itself starts at the beginning of the next line, and is followed by a newline character. It may contain internal newlines. `-aclm <number> <length>`

A line of this form is present for every ACL message variable that is not empty. The number identifies the variable. The length is the length of the data string for the variable. The string itself starts at the beginning of the next line, and is followed by a newline character. It may contain internal newlines. `-active_hostname <hostname>`

This is present if, when the message was received over SMTP, the value of `$smtp_active_hostname` was different to the value of `$primary_hostname`. `-allow_unqualified_recipient`

This is present if unqualified recipient addresses are permitted in header lines (to stop such addresses from being qualified if rewriting occurs at transport time). Local messages that were input using `-bnq` and remote messages from hosts that match `recipient_unqualified_hosts` set this flag. `-allow_unqualified_sender`

This is present if unqualified sender addresses are permitted in header lines (to stop such addresses from being qualified if rewriting occurs at transport time). Local messages that were input using `-bnq` and remote messages from hosts that match `sender_unqualified_hosts` set this flag. `-auth_id <text>`

The id information for a message received on an authenticated SMTP connection &ndash; the value of the `$authenticated_id` variable. `-auth_sender <address>`

The address of an authenticated sender &ndash; the value of the `$authenticated_sender` variable. `-body_linecount <number>`

This records the number of lines in the body of the message, and is always present. `-body_zerocount <number>`

This records the number of binary zero bytes in the body of the message, and is present if the number is greater than zero. `-deliver_firsttime`

This is written when a new message is first added to the spool. When the spool file is updated after a deferral, it is omitted. `-frozen <time>`

The message is frozen, and the freezing happened at `<time>`. `-helo_name <text>`

This records the host name as specified by a remote host in a HELO or EHLO command. `-host_address <address>.<port>`

This records the IP address of the host from which the message was received and the remote port number that was used. It is omitted for locally generated messages. `-host_auth <text>`

If the message was received on an authenticated SMTP connection, this records the name of the authenticator &ndash; the value of the `$sender_host_authenticated` variable. `-host_lookup_failed`

This is present if an attempt to look up the sending host's name from its IP address failed. It corresponds to the `$host_lookup_failed` variable. `-host_name <text>`

This records the name of the remote host from which the message was received, if the host name was looked up from the IP address when the message was being received. It is not present if no reverse lookup was done. `-ident <text>`

For locally submitted messages, this records the login of the originating user, unless it was a trusted user and the `-oMt` option was used to specify an `ident` value. For messages received over TCP/IP, this records the `ident` string supplied by the remote host, if any. `-interface_address <address>.<port>`

This records the IP address of the local interface and the port number through which a message was received from a remote host. It is omitted for locally generated messages. `-local`

The message is from a local sender. `-localerror`

The message is a locally-generated bounce message. `-local_scan <string>`

This records the data string that was returned by the `local_scan()` function when the message was received &ndash; the value of the `$local_scan_data` variable. It is omitted if no data was returned. `-manual_thaw`

The message was frozen but has been thawed manually, that is, by an explicit Exim command rather than via the auto-thaw process. `-N`

A testing delivery process was started using the `-N` option to suppress any actual deliveries, but delivery was deferred. At any further delivery attempts, `-N` is assumed. `-received_protocol`

This records the value of the `$received_protocol` variable, which contains the name of the protocol by which the message was received. `-sender_set_untrusted`

The envelope sender of this message was set by an untrusted local caller (used to ensure that the caller is displayed in queue listings). -spam\_score\_int <number>

If a message was scanned by SpamAssassin, this is present. It records the value of \$spam\_score\_int. -tls\_certificate\_verified

A TLS certificate was received from the client that sent this message, and the certificate was verified by the server. -tls\_cipher <cipher name>

When the message was received over an encrypted connection, this records the name of the cipher suite that was used. -tls\_peerdn <peer DN>

When the message was received over an encrypted connection, and a certificate was received from the client, this records the Distinguished Name from that certificate.

Following the options there is a list of those addresses to which the message is not to be delivered. This set of addresses is initialized from the command line when the -t option is used and extract\_addresses\_remove\_arguments is set; otherwise it starts out empty. Whenever a successful delivery is made, the address is added to this set. The addresses are kept internally as a balanced binary tree, and it is a representation of that tree which is written to the spool file. If an address is expanded via an alias or forward file, the original address is added to the tree when deliveries to all its child addresses are complete.

If the tree is empty, there is a single line in the spool file containing just the text &ldquo;XX&rdquo;. Otherwise, each line consists of two letters, which are either Y or N, followed by an address. The address is the value for the node of the tree, and the letters indicate whether the node has a left branch and/or a right branch attached to it, respectively. If branches exist, they immediately follow. Here is an example of a three-node tree: YY darcy@austen.fict.example  
NN alice@wonderland.fict.example  
NN editor@thesaurus.ref.example

After the non-recipients tree, there is a list of the message's recipients. This is a simple list, preceded by a count. It includes all the original recipients of the message, including those to whom the message has already been delivered. In the simplest case, the list contains one address per line. For example: 4  
editor@thesaurus.ref.example  
darcy@austen.fict.example  
rdo@foundation  
alice@wonderland.fict.example

However, when a child address has been added to the top-level addresses as a result of the use of the one\_time option on a redirect router, each line is of the following form: <top-level address> <errors\_to address> <length>,<parent number>#<flag bits>

The 01 flag bit indicates the presence of the three other fields that follow the top-level address. Other bits may be used in future to support additional fields. The <parent number> is the offset in the recipients list of the original parent of the &ldquo;one time&rdquo; address. The first two fields are the envelope sender that is associated with this address and its length. If the length is zero, there is no special envelope sender (there are then two space characters in the line). A non-empty field can arise from a redirect router that has an errors\_to setting.

A blank line separates the envelope and status information from the headers which follow. A header may occupy several lines of the file, and to save effort when reading it in, each header is preceded by a number and an identifying character. The number is the number of characters in the header, including any embedded newlines and the terminating newline. The character is one of the following: <blank>header in which Exim has no special interestBBcc: headerCCc: headerFFrom: headerIMessage-id: headerPReceived: header &ndash; P for &ldquo;postmark&rdquo;;RReply-To: headerSSender: headerTTo: header\*replaced or deleted header

Deleted or replaced (rewritten) headers remain in the spool file for debugging purposes. They are not transmitted when the message is delivered. Here is a typical set of headers: 11P Received: by hobbit.fict.example with local (Exim 4.00) id 14y9EI-00026G-00; Fri, 11 May 2001 10:28:59 +0100  
049 Message-Id: <E14y9EI-00026G-00@hobbit.fict.example>  
038\* X-rewrote-sender: bb@hobbit.fict.example  
042\* From: Bilbo Baggins <bb@hobbit.fict.example>  
049F From: Bilbo Baggins <B.Baggins@hobbit.fict.example>

099\* To: alice@wonderland.fict.example, rdo@foundation,  
darcy@austen.fict.example, editor@thesaurus.ref.example  
104T To: alice@wonderland.fict.example, rdo@foundation.example,  
darcy@austen.fict.example, editor@thesaurus.ref.example  
038 Date: Fri, 11 May 2001 10:28:59 +0100

The asterisked headers indicate that the envelope sender, From: header, and To: header have been rewritten, the last one because routing expanded the unqualified domain foundation.