

NAME[1]

s-nail [v14.8.0-13-g2fd8894]— send and receive Internet mail

SYNOPSIS[2]

```

s-nail -h | --help
s-nail [-BdEFintv~] [-A account] [-a attachment] [-b bcc-addr] [-c cc-addr]
    [-q quote-file] [-r from-addr] [-S variable[=value]] [-s subject]
    [-X cmd] [-.] to-addr ... [-- mta-option ...]
s-nail [-BdEeHiNnRv~#] [-A account] [-L spec-list] [-r from-addr]
    [-S variable[=value]] [-X cmd] -f [file] [-- mta-option ...]
s-nail [-BdEeHiNnRv~#] [-A account] [-L spec-list] [-r from-addr]
    [-S variable[=value]] [-u user] [-X cmd] [-- mta-option ...]

```

TABLE OF CONTENTS[3]**NAME**[1]**SYNOPSIS**[2]**TABLE OF CONTENTS**[3]**DESCRIPTION**[4]

Options[16]

A starter[17]

Sending mail[18]

Reading mail[19]

Viewing HTML mail and MIME attachments[20]

Mailing lists[21]

Resource files[22]

Character sets[23]

Message states[24]

Specifying messages[25]

On URL syntax and credential lookup[26]

Command line editor[27]

Coloured message display[28]

COMMANDS[5]**TILDE ESCAPES**[6]**VARIABLE OPTIONS**[7]

Initial Settings[29]

Binary options[30]

Value options[31]

ENVIRONMENT[8]**FILES**[9]

The mime.types files[32]

The .netrc file[33]

EXAMPLES[10]

An example configuration[34]

Signed and encrypted messages with S/MIME[35]

Using CRLs with S/MIME or SSL/TLS[36]

Handling spam[37]

SEE ALSO[11]**HISTORY**[12]**AUTHORS**[13]

CAVEATS[14]**BUGS**[15]**DESCRIPTION**[4]

Compatibility note: S-nail and part of its configuration syntax will change in v15.0. Until then there will exist a partial but growing number of backward and forward compatibility configuration options. To choose the new syntax and behaviour already today, the binary option *v15-compat*[340] must be set. The manual will refer to it via [v15-compat] and [no v15-compat] as necessary. Commands and variables which will vanish in v15.0 are marked [Obsolete].

S-nail is a mail processing system with a command syntax reminiscent of *ed*(1) with lines replaced by messages. It is intended to provide the functionality of the POSIX *mailx*(1) command and offers (mostly optional) extensions for line editing, IDNA, MIME, S/MIME, SMTP and POP3. It is usable as a mail batch language.

Options[16]

In the following list of supported command line options, **-d** [43], **-E** [44], **-i** [50], **-N** [52] and **-v** [62] are implemented by means of setting the respective option, as via **-S** [57]. [*mta-option* . . .] arguments that are given at the end of the command line after a ‘--’ separator will be passed through to the mail-transfer-agent (MTA) and persist for an entire (interactive) session – if the setting of *expandargv*[365] allows their recognition; MTA arguments can also be specified in the variable *sendmail-arguments*[409]; find MTA interaction described in more detail in the documentation of *sendmail*[408]. MTA arguments are ignored when mail is send via SMTP data transfer.

- A** *account* Executes an **account** [76] command (see below) for *account* after the startup files have been read.
- a** *file* Attach the given file to the message. The same filename conventions as described in the section **COMMANDS**[5] apply: shell word expansion is restricted to the tilde ‘~’ and variables.
- B** Make standard input and standard output line-buffered.
- b** *address* Send blind carbon copies to the given list of addresses. **Sending mail** [18] below goes into more detail on that.
- c** *address* Send carbon copies to the given list of addresses.
- d** Set the *debug*[284] variable, which enables debug messages and disables message delivery.
- E** Set the *skipemptybody*[332] variable and thus discard messages with an empty message part body. This is useful for sending messages from scripts.
- e** Just check if mail is present in the system mailbox. If yes, return an exit status of zero, a non-zero value otherwise.
- F** Save the message to send in a file named after the local part of the first recipient’s address.
- f** [*file*] Read in the contents of the user’s MBOX [470] (or the specified file) for processing; when S-nail is quit, it writes undeleted messages back to this file (but be aware of the *hold*[297] option). Some special conventions are recognized for the string *file* which are documented for the **file**[105] command below. Note that *file* is not a direct argument to the flag **-f** [47], but is instead taken from the command line after option processing has been completed. In order to use a *file* that starts with a hyphen, prefix it with a (relative) path, as in *./-hyphenbox.mbox*.

- H** Print a header summary of all messages and exit. A configurable summary view is available via the **-L** [51] option.
- h** Print a short usage summary. Because of widespread use a **--help** argument will have the same effect.
- i** Set the *ignore*[299] variable to ignore tty interrupt signals.
- L *spec-list*** Print a header summary of only those messages that match the given *spec-list*, then exit. See the section **Specifying messages** [25] for the format of *spec-list*. If the **-H**[48] option has been given in addition no header summary is produced, but S-nail will instead indicate via its exit status whether *spec-list* matched any messages ('0') or not ('1'); note that any verbose output is suppressed in this mode and must instead be enabled explicitly (e.g., by using the option **-v** [62]).
- N** Unset the *header*[294] variable and thus inhibit initial display of message headers when reading mail or editing a mail folder.
- n** Inhibits reading */usr/local/etc/s-nail.rc* [67] upon startup. This option should be activated for S-nail scripts that are invoked on more than one machine, because the contents of that file may differ between them. (The same behaviour can be achieved by setting the *NAIL_NO_SYSTEM_RC* [472] environment variable.)
- q *file*** Start the message with the contents of the specified file. May be given in send mode only.
- R** Any folder opened will be in read-only mode.
- r *address*** If *address* is a valid address then it specifies the envelope sender address to be passed to the MTA as *'-f address'* when a message is sent. Shall *address* include a user name, comments etc., then the components will be separated and the name part will be passed to the MTA individually via *'-F name'*. The given *address* will also be assigned to the *from*[371] variable, just as if additionally *-sfrom=address* had been specified (therefore affecting SMTP data transfer, too).

If instead an empty string is passed as *address* then the content of the variable *from*[371] will be evaluated and used for this purpose whenever the MTA is contacted. Note that S-nail by default, without **-r** that is, neither passes *'-f'* nor *'-F'* flags to the MTA by itself.
- s *variable*[=value]** Sets the internal option *variable* and, in case of a value option, assigns *value* to it. Even though options set via **-s** [57] may be overwritten from within resource files, the command line setting will be reestablished after all resource files have been loaded.
- s *subject*** Specify the subject of the to-be-sent message.
- t** The message to be sent is expected to contain a message header with *To:*, *Cc:*, or *Bcc:* fields giving its recipients and *Subject:* giving the subject of the message. Recipients and subject specified on the command line are ignored.
- u *user*** Read the system mailbox of *user* (appropriate privileges presumed), and "assume to be" *user* in some aspects, e.g. in respect to **file**[105]-expansions of *'%'* etc.; also see **USER**[480].
- v** Print S-nail's version and exit.

- v** Setting the *verbose*[341] option causes some verbosity (like printing of certificate chains). Using it twice increases the level of verbosity.
- x cmd** (EXPERIMENTAL) Add the given *cmd* to the list of commands to be executed before normal operation starts. Correlates with **-#**[65] and *batch-exit-on-error*[274]; the only possibility to execute commands in non-interactive mode when reading startup files is actively prohibited.
- ~** Enable **TILDE ESCAPES**[6] even if not in interactive mode.
- #** This sets several options to prepare S-nail for working in (most likely non-interactive) batch mode: *dot*[286], *emptystart*[289], *noheader*[294], *quiet*[319], *sendwait*[328], as well as *MBOX*[470] and *folder*[367] (both to /dev/null). It also enables processing of **TILDE ESCAPES**[6]. E.g., the following should send an email message to “bob”:


```
$ LC_ALL=C printf 'm bob\n~s subject\nText\n.\nx\n' | \
  LC_ALL=C MAILRC=/dev/null s-nail -n -#
```
- .** This flag forces termination of option processing in order to address “option injection”. It also forcefully puts S-nail into send mode, see **Sending mail**[18].

A starter[17]

S-nail is a direct descendant of the BSD Mail program that was introduced in 1978 (itself superceding the simpler UNIX mail program) and used to introduce itself (in the Mail reference manual) as follows:

Mail provides a simple and friendly environment for sending and receiving mail. It divides incoming mail into its constituent messages and allows the user to deal with them in any order. In addition, it provides a set of ed(1)-like commands for manipulating messages and sending mail. Mail offers the user simple editing capabilities to ease the composition of outgoing messages, as well as providing the ability to define and send to names which address groups of users.

S-nail is thus the user side of the Unix mail system, whereas the system side (mail-transfer-agent, MTA) was traditionally taken by *sendmail*(8); today *postfix*(1) or *exim*(8) are often used for this purpose instead. If the [Option]al SMTP feature has been built into S-nail then the system side is not a mandatory precondition for mail delivery.

Because S-nail strives for compliance with POSIX *mailx*(1) it is likely that some configuration settings have to be adjusted before using it is a smooth experience. The default global */usr/local/etc/s-nail.rc*[67] file already bends those standard settings a bit towards more user friendliness and safety, e.g., it sets the *hold*[297] and *keepsave*[302] options in order to suppress the automatic moving of messages to *MBOX*[470] that would otherwise occur (see **Message states**[24]) and *keep*[301] to not remove empty files in order not to mangle file permissions when files eventually get recreated. It does not set the *folder* [367] option so that by default file grouping (via the ‘+’ prefix as documented also for *file*[105]) is not enabled. The section **EXAMPLES**[10] contains some further suggestions.

Sending mail[18]

To send a message to one or more people, using a local mail-transfer-agent (MTA; the executable path can be set via *sendmail*[408]) or the [Option]al builtin SMTP (set and see the variable *smtp*[422]) transport to actually deliver the generated mail message, S-nail can be invoked with arguments which are the names of people to whom the mail will be sent:

```
$ s-nail -s Subject -a attachm.ent bill@host1 'Bob <bob@host2>'
# But... try it in an isolated (almost) dry-run mode first
$ LC_ALL=C MAILRC=/dev/null s-nail -n -dvv \
  -Ssmtp -Sfrom="me <he@re>" -Srecord=./sent-mails.mbox \
  -s Subject "(Lovely) Bob <bob@host2>"
```

```
$ cat ./sent-mails.mbox
```

The user is then expected to type in the message contents. In this compose mode S-nail treats lines beginning with the character ‘~’ special – these are so-called **TILDE ESCAPES** [6] which can be used to read in files, process shell commands, add and edit attachments and more; e.g., the tilde escape **~e** [240] will start the text editor to revise the message in it’s current state, **~h** [244] allows editing of the message recipients and **~?** [231] gives an overview of available tilde escapes. Typing **control-D** (‘^D’) at the beginning of an empty line leaves compose mode and causes the message to be send, whereas typing **control-C** (‘^C’) twice will abort the current letter (saving the contents which yet have been collected in **DEAD** [460]).

A number of options can be used to alter default behavior; e.g., **set** [173]ing (also via **-s** [57]) *editalong* [287] will automatically startup a text editor when compose mode is entered, *askcc* [267] will cause the user to be prompted actively for carbon-copy recipients and the *dot* [286] option will allow leaving compose mode by writing a line consisting solely of a dot (‘.’).

Very important, though, is to define which **Character sets** [23] may be used when sending messages, usually by setting the option *sendcharsets* [406] accordingly, having read the section **The mime.types files** [32] to understand how the MIME-type of outgoing attachments is classified and the knowledge that messages are sent asynchronously unless *sendwait* [328] is set: only with it MTA delivery errors will be recognizable.

Saving a copy of the sent messages in a *record* [401] may also be desirable – as for most mailbox file targets some special conventions are recognized, see the **file** [105] command for more on that. The section **EXAMPLES** [10] contains an example configuration for sending messages via some of the well-known public mail providers.

Message recipients (as specified on the command line or defined in **To:**, **Cc:** or **Bcc:**) may not only be email addressees but can also be names of mailboxes and even complete shell command pipe specifications. Proper quoting may be necessary, e.g., to embed whitespace characters. If the variable *expandaddr* [364] is not set then only network addresses (see *mailaddr*(7) for a description of mail addresses) and aliases may be used, though.

If the variable *expandaddr* [364] is set the following rules are used to classify recipient addresses: Any name which starts with a vertical bar ‘|’ character specifies a pipe – the command string following the ‘|’ is executed and the message is sent to its standard input; Any other name which contains an at sign ‘@’ character is treated as a network address; Any other name which starts with a plus sign ‘+’ character specifies a mailbox name; Any other name which contains a slash ‘/’ character but no exclamation mark ‘!’ or percent sign ‘%’ character before also specifies a mailbox name; What remains is treated as a network address.

```
$ echo bla | s-nail -Sexpandaddr -s test ./mbox.mbox
$ echo bla | s-nail -Sexpandaddr -s test '|cat >> ./mbox.mbox'
$ echo safe | LC_ALL=C MAILRC=/dev/null s-nail -n \
  -Sexpandaddr=fail,noalias -s test -. bob@home.net
```

It is possible to create personal distribution lists via the **alias** [77] command, so that, for instance, the user can send mail to **cohorts** and have it go to a group of people:

```
alias cohorts bill jkf mark kridle@ucbcory ~/mail/cohorts.mbox
```

Please note that this mechanism has nothing in common with the system wide aliases that may be used by the local MTA (mail-transfer-agent), which are subject to the “noalias” constraint of *expandaddr* [364] and are often tracked in a file `/etc/aliases` (and documented in *aliases*(5) and *sendmail*(1)). Personal aliases will be expanded by S-nail before the message is sent, and are thus a convenient alternative to specifying each addressee by itself.

To avoid environmental noise scripts should “detach” s-nail from any configuration files and create a script-local environment, either by pointing the **MAILRC** [471] variable to a custom configuration file, by passing the configuration in environment variables, or by using the **-s** [57] command line option to specify options:

```
$ env LC_ALL=C MAILRC=/dev/null password=SECRET s-nail -n \
  -Sv15-compat \
  -S 'smtp=smtps://mylogin@some.host:465' -Ssmtp-auth=login \
  -S 'from=scriptreply@domain' \
  -s 'subject' -a attachment_file \
  "Recipient 1 <recipient1@domain>" recipient2@domain \
  < content_file
```

Reading mail[19]

When invoked without addressees S-nail enters interactive mode in which mails may be read. When used like that the user's system mailbox is read in and a one line header of each message therein is printed. Note that if the mailbox is empty S-nail will exit after printing a message unless the option *emptystart*[289] is found to be set.

At the *prompt*[398] the command **list**[127] will give a listing of all available commands and **help**[122] will give a summary of the most useful ones. If the [Option]al documentation strings are available one can type '?X' and see the actual expansion of 'X' and what it's purpose is, i.e., commands can be abbreviated (note that POSIX defines some abbreviations, so that the alphabetical order of commands doesn't necessarily relate to the abbreviations; it is possible to define overwrites with the **ghost**[120] command, however).

Messages are given numbers (starting at 1) which uniquely identify messages; the current message – the “dot” – will either be the first new message, or the first unread message, or the first message of the mailbox; the option *showlast*[329] will instead cause usage of the last message for this purpose.

Messages can be printed with the **print**[150] command, or short: 'p'. By default the current message (“dot”) is printed, but like with most other commands it is possible to give a fancy message specification (see **Specifying messages**[25]), e.g., p:u will display all unread messages, 'p.' will print the “dot”, p 1 5 will print the messages 1 and 5, p 1-5 will print the messages 1 through 5, and 'p-' and 'p+' will print the last and the next message, respectively.

In the default setup all header fields of a message will be printed, but this can be changed: either by blacklisting a list of fields via **ignore**[126], or by whitelisting only a given list with the **retain**[166] command, e.g., **retain** date from to cc subject. In order to print all header fields of a message regardless of currently active ignore or retain lists, use the command **Print**[149].

Dependent upon the configuration a **Command line editor**[27] aims at making user experience with the many **COMMANDS**[5] a bit nicer. When reading the system mailbox or when **-f** [47] (or **file**[105]) specified a mailbox explicitly prefixed with the special '%:' modifier then messages which have been read will be moved to the user's **MBOX**[470] file automatically when the mailbox is left, either by changing the active mailbox or by quitting S-nail (also see **Message states**[24]).

After examining a message the user can also **delete**[92] 'd' the message, **reply** [157] 'r' to the sender and all recipients or **Reply**[156] 'R' exclusively to the sender. Deletion causes S-nail to forget about the message; This is not irreversible, though, one can **undelete**[197] 'u' the message by giving its number, or the S-nail session can be ended by giving the **exit**[102] 'x' command.

To end a mail processing session one may either issue **quit**[151] 'q' to cause a full program exit, which possibly includes automatic moving of read messages to **MBOX**[470] as well as updating the [Option]al command line editor history file, or use the command **exit**[102] 'x' instead in order to prevent any of these actions.

Viewing HTML mail and MIME attachments[20]

Messages which are HTML-only get more and more common and of course many messages come bundled with a bouquet of MIME attachments. Whereas S-nail [Option]ally supports a simple HTML-to-text con-

verter to deal with HTML messages (see **The mime.types files**[32]), it normally can't deal with any of these itself, but instead programs need to become registered to deal with specific MIME types or file extensions. These programs may either prepare plain text versions of their input in order to enable S-nail to display the content on the terminal, or display the content themselves, for example in a graphical window. The latter type of programs by default “blocks” S-nail until the external viewer has terminated, but asynchronous side-by-side execution is also possible, in which case S-nail will continue to display the message and remain responsive.

To install an external handler program for a specific MIME type set an according *pipe-TYPE/SUBTYPE*[390] variable. To define a handler for a specific file extension set the respective *pipe-EXTENSION*[396] variable – these handlers take precedence. The variable *mime-counter-evidence*[379] can be set to improve dealing with faulty MIME part declarations as are often seen in real-life messages. E.g., to display a HTML message inline (that is, converted to a more fancy plain text representation than the builtin converter is capable to produce) with either of the text-mode browsers *lynx*(1) or *elinks*(1), teach S-nail about MathML documents and make it display them as plain text and to open PDF attachments in an external PDF viewer, asynchronously:

```
if $features !@ HTML-FILTER
  #set pipe-text/html="elinks -force-html -dump 1"
  set pipe-text/html="lynx -stdin -dump -force_html"
  # Display HTML as plain text instead
  #set pipe-text/html=@
endif
mimetype '@ application/mathml+xml mathml'
set pipe-application/pdf="@&cat > \
  \"${NAIL_TMPDIR}/${NAIL_FILENAME}\"; \
  acroread \"${NAIL_TMPDIR}/${NAIL_FILENAME}\"; \
  rm \"${NAIL_TMPDIR}/${NAIL_FILENAME}\""
```

Note: special care must be taken when using such commands as mail viruses may be distributed by this method: if messages of type *application/x-sh* or files with the extension *.sh* were blindly filtered through the shell, for example, a message sender could easily execute arbitrary code on the system S-nail is running on. For more on MIME, also in respect to sending of messages, see the section **The mime.types files**[32] and the command **mimetype**[133].

Mailing lists[21]

S-nail offers some support to ease handling of mailing lists. The command **mlist**[134] promotes all given arguments to known mailing lists, and **mmlsubscribe** [135] sets their subscription attribute, creating them first as necessary. (On the other hand **unmmlsubscribe** [206] doesn't **unmmlist**[205] automatically, but only resets the subscription attribute.) Using the commands without arguments will print out (a subset of) all currently defined mailing lists. The *headline*[373] format ‘%T’ can be used to mark out messages with configured list addresses in the header display.

[Option]ally mailing lists may also be specified as regular expressions, which allows matching of many addresses with a single expression. However, all fully qualified list addresses are matched via a fast dictionary, whereas expressions are placed in (a) list(s) which is (are) matched sequentially.

```
set followup-to followup-to-honour=ask-yes reply-to-honour=ask-yes
mlist a1@b1.c1 a2@b2.c2 .*@lists.c3$
mmlsubscribe a4@b4.c4 exact@lists.c3
```

The variable *followup-to-honour*[370] will ensure that a *Mail-Followup-To:* header is honoured when the message is being replied to (via **reply**[157] and **Lreply**[129]) and *followup-to*[291] controls whether this header is created when sending mails; it will be created automatically for a couple of reasons, too, like when the special “mailing list specific” respond command **Lreply**[129] is used, when

reply[157] is used to respond to a message with its `Mail-Followup-To:` being honoured etc.

A difference in between the handling of known and subscribed lists is that the address of the sender is usually not part of a generated `Mail-Followup-To:` when addressing the latter, whereas it is for the former kind of lists. Usually because there are exceptions: say, if multiple lists are addressed and not all of them are subscribed lists.

For convenience S-nail will, temporarily, automatically add a list address that is presented in the `List-To:` header of a message that is being responded to to the list of known mailing lists. Shall that header have existed S-nail will instead, dependend on the variable *reply-to-honour*[404], use an also set `Reply-To:` for this purpose in order to accept a list administrators' wish that is supposed to have been manifested like that (but only if it provides a single address which resides on the same domain as what is stated in `List-To:`).

Resource files[22]

Upon startup S-nail reads in several resource files:

`/usr/local/etc/s-nail.rc`

System wide initialization file. Reading of this file can be suppressed, either by using the `-n`[53] command line option, or by setting the environment variable `NAIL_NO_SYSTEM_RC`[472].

`~/mailrc`

File giving initial commands. A different file can be chosen by setting the environment variable `MAILRC`[471].

`NAIL_EXTRA_RC`[381]

Can be used to define an optional startup file to be read after `~/mailrc`[68]. This variable is only honoured in certain circumstances (see its documentation for more).

The content of these files is interpreted as follows:

- A lines' leading whitespace is ignored.
- An empty line is ignored.
- If the line (content) starts with the number sign '#' then it is a comment-command – a real command! – and also ignored. This command is the only form of comment that is understood.
- A line can be “continued” on the next line if the newline character is “escaped” by preceding it with the backslash character '\'. Note that any leading whitespace of follow lines is removed: If whitespace is desired it must be placed before the backslash. For example:

```
# This line is a comment command. And y\
  es, it is really continued here.
set debug \
  verbose
```

Character sets[23]

[Option] S-nail detects the character set of the terminal by using mechanisms that are controlled by the `LC_CTYPE`[466] locale setting (the manual for `setlocale(3)` should give an overview); the S-nail internal variable *tycharset*[453] will be set to the detected terminal character set accordingly and will thus show up in the output of the commands `set`[173] and `varshow`[220].

However, a user supplied *tycharset*[453] value is not overwritten by this detection mechanism: this “feature” must be used if the detection doesn't work properly, and it may be used to adjust the name of the locale character set. E.g., on BSD systems one may use a locale with the character set `ISO8859-1`, which is not a valid name for this character set; to be on the safe side, one may set *tycharset*[453] to the correct name, which is `ISO-8859-1`.

Note that changing the value doesn't mean much beside that, since several aspects of the real character set are implied by the locale environment of the system, and that stays unaffected by the content of an overwritten *tycharset*[453] variable. (This is mostly an issue when interactively using S-nail, though. It is actually possible to send mail in a completely “faked” locale environment.)

If no character set conversion capabilities have been compiled into S-nail (i.e., no *iconv*(3) library has been found), then *tycharset*[453] will be the only supported character set, it is simply assumed that it can be used to exchange 8-bit messages, and the rest of this section does not apply; it may however still be necessary to explicitly set it if automatic detection fails, since in that case it defaults to the mentioned ISO-8859-1.

When reading messages, their text is converted into *tycharset*[453] as necessary in order to display them on the users terminal. Unprintable characters and invalid byte sequences are detected and replaced by proper substitution characters (unless the variable *print-all-chars*[317] was set once S-nail was started). Also see *charset-unknown-8bit*[350] to deal with another hairy aspect of message interpretation.

When sending messages all their parts and attachments are classified. Whereas no character set conversion is performed on those parts which appear to be binary data, the character set being used must be declared within the MIME header of an outgoing text part if it contains characters that do not conform to the set of characters that are allowed by the email standards. Permissible values for character sets can be declared using the *sendcharsets*[406] variable, and *charset-8bit*[349], which defines a catch-all last-resort fallback character set that is implicitly appended to the list of character-sets in *sendcharsets*[406].

When replying to a message and the variable *reply-in-same-charset*[323] is set then the character set of the message being replied to is tried first. And it is also possible to make S-nail work even more closely related to the current locale setting automatically by using the variable *sendcharsets-else-tycharset*[327], please see there for more information.

All the specified character sets are tried in order unless the conversion of the part or attachment succeeds. If none of the tried (8-bit) character sets is capable to represent the content of the part or attachment, then the message will not be sent and its text will be saved to *DEAD*[460]. In general, if the message “Cannot convert from a to b” appears, either some characters are not appropriate for the currently selected (terminal) character set, or the needed conversion is not supported by the system. In the first case, it is necessary to set an appropriate *LC_CTYPE*[466] locale and/or the variable *tycharset*[453].

The best results are usually achieved when S-nail is run in a UTF-8 locale on a UTF-8 capable terminal, in which case the full Unicode spectrum of characters is available. In this setup characters from various countries can be displayed, while it is still possible to use more simple character sets for sending to retain maximum compatibility with older mail clients.

Message states[24]

S-nail differentiates in between several different message states; the current state will be reflected in the header display if *headline*[373] is configured to do so. In S-nail message states are inspected when leaving a mailbox and may cause messages to be automatically moved to the special *MBOX*[470] mailbox – because this may be irritating to users which are used to “more modern” mail-user-agents, the default global */usr/local/etc/s-nail.rc*[67] sets the *hold*[297] and *keepsave*[302] variables in order to suppress this behaviour.

new	Message has neither been viewed nor moved to any other state. Such messages are retained even in the system mailbox.
unread	Message has neither been viewed nor moved to any other state, but the message was present already when the mailbox has been opened last: Such messages are retained even in the system mailbox.

- read** The message has been processed by one of the following commands: `~f[242]`, `~m[247]`, `~F[241]`, `~M[246]`, `copy[87]`, `mbox[132]`, `next[142]`, `pipe[147]`, `print[150]`, `Print[149]`, `top[188]`, `type[191]`, `Type[190]`, `undelete[197]`. The `delete[92]`, `dp[94]`, and `dt[95]` commands may also cause the next message to be marked as read, depending on the value of the `autoprint[271]` variable. Except when the `exit[102]` command is used, messages that are in the system mailbox or in mailboxes which were opened with the special ‘%:’ prefix and are in read state when the mailbox is left will be saved in `MBOX[470]` unless the option `hold[297]` is set.
- deleted** The message has been processed by one of the following commands: `delete[92]`, `dp[94]`, `dt[95]`. Only `undelete[197]` can be used to access such messages.
- preserved** The message has been processed by a `preserve[148]` command and it will be retained in its current location.
- saved** The message has been processed by one of the following commands: `save[168]` or `write[223]`. Unless when the `exit[102]` command is used, messages that are in the system mailbox or in mailboxes which were opened with the special ‘%:’ prefix and are in saved state when the mailbox is left will be deleted; they will be saved in `MBOX[470]` when the option `keepsave[302]` is set.

Specifying messages[25]

Commands such as `print[150]` and `delete[92]` can be given a list of message numbers as arguments to apply to a number of messages at once. Thus `delete 1 2` deletes messages 1 and 2, whereas `delete 1-5` will delete the messages 1 through 5. In sorted or threaded mode (see the `sort[179]` command), `delete 1-5` will delete the messages that are located between (and including) messages 1 through 5 in the sorted/threaded order, as shown in the header summary. Multiple colon modifiers can be joined into one, e.g., `:du`. The following special message names exist:

- `:n` All new messages.
- `:o` All old messages (any not in state read or new).
- `:u` All unread messages.
- `:d` All deleted messages (for the `undelete[197]` command).
- `:r` All read messages.
- `:f` All `flag[106]`ged messages.
- `:a` All answered messages (cf. the `markanswered[304]` variable).
- `:t` All messages marked as draft.
- `:s` [Option] All messages classified as spam.
- `:S` [Option] All messages with unsure spam classification.
- `.` The current message, the so-called “dot”.
- `;` The message that was previously the current message.
- `,` The parent message of the current message, that is the message with the Message-ID given in the `In-Reply-To:` field or the last entry of the `References:` field of the current message.
- `-` The next previous undeleted message, or the next previous deleted message for the `undelete[197]` command. In sorted/threaded mode, the next previous such message in the sorted/threaded order.
- `+` The next undeleted message, or the next deleted message for the `undelete[197]` command. In sorted/threaded mode, the next such message in the sorted/threaded order.
- `^` The first undeleted message, or the first deleted message for the `undelete[197]` command. In sorted/threaded mode, the first such message in the sorted/threaded order.
- `$` The last message. In sorted/threaded mode, the last message in the sorted/threaded order.

&x In threaded mode, selects the message addressed with *x*, where *x* is any other message specification, and all messages from the thread that begins at it. Otherwise it is identical to *x*. If *x* is omitted, the thread beginning with the current message is selected.

*** All messages.

' All messages that were included in the message list for the previous command.

/string

All messages that contain *string* in the subject field (case ignored). See also the *searchheaders*[326] variable. If *string* is empty, the string from the previous specification of that type is used again.

[@name-list]@expr

All messages that contain the given case-insensitive search *expression*; if the [Option]al regular expression (see *re_format*(7)) support is available *expr* will be interpreted as one if any of the “magical” regular expression characters is seen. If the optional *@name-list* part is missing, the search is restricted to the subject field body, but otherwise *name-list* specifies a comma-separated list of header fields to search, as in

```
'@to,from,cc@Someone i ought to know'
```

The special names *header* or *'<'* can be used to search in the header of the message, and the special names *body* or *'>'* and *text* or *'='* can be used to perform full text searches – whereas the former searches only the body, the latter also searches the message header. In order to search for a string that includes a *'@'* (commercial at) character the *name-list* is effectively non-optional, but may be given as the empty string.

address

All messages from *address*. By default, this is a case-sensitive search for the complete email address. If the *allnet* [261] variable is set, only the local part of the addresses is evaluated for the comparison. Otherwise, if the *showname* [330] variable is set, a case-sensitive search for the complete real name of a sender is performed. The IMAP-style (*from address*) expression can be used instead if substring matches are desired.

[Option] IMAP-style SEARCH expressions may also be used. This addressing mode is available with all types of folders; S-nail will perform the search locally as necessary. Strings must be enclosed by double quotes *'"* in their entirety if they contain white space or parentheses; within the quotes, only backslash *'\'* is recognized as an escape character. All string searches are case-insensitive. When the description indicates that the “envelope” representation of an address field is used, this means that the search string is checked against both a list constructed as

```
("name" "source" "local-part" "domain-part")
```

for each address, and the addresses without real names from the respective header field. These search expressions can be nested using parentheses, see below for examples.

(criterion)

All messages that satisfy the given *criterion*.

(criterion1 criterion2 ... criterionN)

All messages that satisfy all of the given criteria.

(or criterion1 criterion2)

All messages that satisfy either *criterion1* or *criterion2*, or both. To connect more than two criteria using *'or'* specifications have to be nested using additional parentheses, as with *(or a (or b c))*, since *(or a b c)* really means *((a or b) and c)*. For a simple *'or'* operation of independent criteria on the lowest nesting level, it is possible to achieve similar effects by using three separate criteria, as with *(a) (b) (c)*.

- (*not criterion*)
All messages that do not satisfy *criterion*.
- (*bcc "string"*)
All messages that contain *string* in the envelope representation of the Bcc: field.
- (*cc "string"*)
All messages that contain *string* in the envelope representation of the Cc: field.
- (*from "string"*)
All messages that contain *string* in the envelope representation of the From: field.
- (*subject "string"*)
All messages that contain *string* in the Subject: field.
- (*to "string"*)
All messages that contain *string* in the envelope representation of the To: field.
- (*header name "string"*)
All messages that contain *string* in the specified Name: field.
- (*body "string"*)
All messages that contain *string* in their body.
- (*text "string"*)
All messages that contain *string* in their header or body.
- (*larger size*)
All messages that are larger than *size* (in bytes).
- (*smaller size*)
All messages that are smaller than *size* (in bytes).
- (*before date*)
All messages that were received before *date*, which must be in the form d[d]-mon-yyyy, where 'd' denotes the day of the month as one or two digits, mon is the name of the month – one of Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec, and yyyy is the year as four digits, e.g., 28-Dec-2012.
- (*on date*)
All messages that were received on the specified date.
- (*since date*)
All messages that were received since the specified date.
- (*sentbefore date*)
All messages that were sent on the specified date.
- (*senton date*)
All messages that were sent on the specified date.
- (*sentsince date*)
All messages that were sent since the specified date.
- () The same criterion as for the previous search. This specification cannot be used as part of another criterion. If the previous command line contained more than one independent criterion then the last of those criteria is used.

On URL syntax and credential lookup[26]

[v15-compatible] For accessing protocol-specific resources, like a POP3 mailbox, usage of compact and standardized Uniform Resource Locators (URL, RFC 1738) has become omnipresent. S-nail expects and understands URLs in the following form; parts in brackets '[']' denote optional parts, optional either because there also exist other ways to define the information in question or because support of the part is protocol-specific – e.g., /path is used by the IMAP protocol but not by POP3.

```
PROTOCOL://[USER[:PASSWORD]@]server[:port][[/path]]
```

If USER and PASSWORD are specified as part of an URL they must be given in URL percent encoded (RFC 3986) form – the command `urleencode` [218] can be used to perform the encoding and show the encoded value. (This doesn't really conform to any standard, but for one it isn't used for any data exchange over the

internet, and second it's easier for users to simply call `urlencode[218]` on a string and use that instead of having to deal with several different standards.) On the other hand, values given in variables are expected not to be URL percent encoded.

Many variable options of S-nail exist in multiple versions, called variable chains for the rest of this document: the plain `variable` as well as `variable-HOST` and `variable-USER@HOST`. Here `HOST` indeed means `server:port` if a `port` had been specified in the respective URL, otherwise it refers to the plain server. Also, `USER` isn't truly the `USER` that had been found when doing the user chain lookup as is described below, i.e., this `USER` will never be in URL percent encoded form, whether it came from an URL or not.

For example, whether an hypothetical URL `smtp://hey%3Ayou@our.house` had been given that includes a user, or whether the URL was `smtp://our.house` and the user had been found differently, to lookup the variable chain `smtp-use-starttls[336]` S-nail first looks for whether `smtp-use-starttls-hey:you@our.house` is defined, then whether `smtp-use-starttls-our.house` exists before finally ending up looking at the plain variable itself.

S-nail obeys the following logic scheme when dealing with the necessary credential informations of an account:

- If no `USER` has been given in the URL the variables `user-HOST` and `user[454]` are looked up; if no such variable(s) can be found then S-nail will, when enforced by the [Option]al variables `netrc-lookup-HOST` or `netrc-lookup[309]`, search the users `.netrc` file for a `HOST` specific entry which provides a `login` name: this lookup will only succeed if unambiguous (one possible matching entry for `HOST`).

If there is still no `USER` then S-nail will fall back to the user who is supposed to run S-nail: either the name that has been given with the `-u[60]` command line option (or, equivalently, but with less precedence, the environment variable `USER[480]`), or `getpwnid(getuid())` a.k.a. the current login user. The identity of this user has been fixated during S-nail startup and is known to be a valid user on the current host.

- Authentication: unless otherwise noted this will lookup the `PROTOCOL-auth-USER@HOST`, `PROTOCOL-auth-HOST`, `PROTOCOL-auth` variable chain, falling back to a protocol-specific default should this have no success.
- If no `PASSWORD` has been given in the URL – it should be noted once that specifying the password in the URL is only syntactic sugar for the user, it'll never be part of an URL that S-nail uses itself –, then if the `USER` has been found through the [Option]al `.netrc` file lookup then that may have already provided the password, too. Otherwise the variable chain `password-USER@HOST`, `password-HOST`, `password[389]` is looked up.

[Option] Then if any of the variables of the chain `agent-shell-lookup-USER@HOST`, `agent-shell-lookup-HOST`, `agent-shell-lookup[343]` is set the shell command specified therein is run and the output (less newline characters) will be used as the password. It is perfectly valid for such an agent to simply not return any data, in which case the password lookup is continued somewhere else; Any command failure is treated as a hard error, however.

The next variable chain that is inspected is the [Option]al `netrc-lookup-USER@HOST`, `netrc-lookup-HOST`, `netrc-lookup[309]`, but this time looking only for the password (multiple user accounts for a single machine may exist as well as a fallback entry without user but with a password).

If at that point there is still no password available, but the (protocols') chosen authentication type requires a password, then in interactive mode the user will be prompted on the terminal.

Note: S/MIME verification works relative to the values found in the From: (or Sender:) header field(s), which means that the values of *smime-sign*[334], *smime-sign-cert*[420] and *smime-sign-include-certs*[421] will not be looked up using the USER and HOST chains from above but instead use the corresponding values from the message that is being worked on. In unusual cases multiple and different USER and HOST combinations may therefore be involved – on the other hand those unusual cases become possible. The usual case is as short as:

```
set smtp=smtp://USER:PASS@HOST smtp-use-starttls \
    smime-sign smime-sign-cert=+smime.pair
```

The section **EXAMPLES**[10] contains complete example configurations.

Command line editor[27]

[Option] S-nail can be configured to support a command line editor and command history lists which are saved in between sessions. One may link against fully-fledged external libraries (*readline*(6), *editline*(3)) or use S-nail's own command line editor NCL (Nail-Command-Line) instead, which should work in all environments which comply to the ISO C standard (ISO/IEC 9899:1990/Amendment 1:1995). When an external library is used, interactive behaviour of S-nail relies on that library and may not correspond one-to-one to what is described in this manual.

Regardless of the actually used command line editor history entries will be created for lines entered in command mode only, and creation of such an entry can be forcefully suppressed by starting the line with a space character. Note that history handling is by itself an optional feature and may therefore not be available. For more information see the documentation of the options *emptystart*[289], *history-gabby*[295], *history-gabby-persist*[296], *line-editor-disable*[303], *NAIL_HISTFILE*[383] and *NAIL_HISTSIZ*E[384].

The builtin S-nail command line editor supports the following operations; the notation *^*-character stands for the combination of the “Control” key plus the mentioned character, e.g., ‘*^*A’ means “hold down control key and press the A key”:

- ‘*^*A’ Go to the start of the line.
- ‘*^*B’ Move the cursor backward one character.
- ‘*^*D’ Forward delete the character under the cursor; quits S-nail if used on the empty line unless the *ignoreeof*[300] option is set.
- ‘*^*E’ Go to the end of the line.
- ‘*^*F’ Move the cursor forward one character.
- ‘*^*G’ Cancel current operation, full reset. If there is an active history search or tabulator expansion then this command will first reset that, reverting to the former line content; thus a second reset is needed for a full reset in this case. In all cases S-nail will reset a possibly used multibyte character input state machine.
- ‘*^*H’ The same as “backspace”: backward delete one character.
- ‘*^*I’ [Option] The same as “horizontal tabulator”: try to expand the word before the cursor. (Note this is not “tabulator-completion” as is known from the *sh*(1) but really means the usual S-nail expansion, as documented for *file*[105], yet it involves shell expansion as a last step, too.)
- ‘*^*J’ The same as “ENTER”: complete this line of input.
- ‘*^*K’ Delete all characters from the cursor to the end of the line.
- ‘*^*L’ Repaint the line.
- ‘*^*N’ [Option] Go to the next history entry.
- ‘*^*O’ Execute the command *dp*[94].
- ‘*^*P’ [Option] Go to the previous history entry.
- ‘*^*R’ [Option] Complete the current line from (the remaining older) history entries.
- ‘*^*U’ The same as ‘*^*A’ followed by ‘*^*K’.

- '^W' Delete the characters from the one preceding the cursor to the preceding word boundary.
- '^X' Move the cursor forward one word boundary.
- '^Y' Move the cursor backward one word boundary.

If problems with commands that are based upon rightwise movement are encountered, adjustments of the option *line-editor-cursor-right*[377] may solve the problem, as documented for it.

If the terminal produces key sequences which are compatible with *xterm*(1) then the left and right cursor keys will map to '^B' and '^F', respectively, the up and down cursor keys will map to '^P' and '^N', and the Home/End/PgUp/PgDown keys will call the **z**[225] command with the respective arguments '0', '\$', '-' and '+' (i.e., perform scrolling through the header summary list).

Coloured message display[28]

[Option] S-nail can be configured to support coloured message display, realized by emitting ANSI colour escape sequences. Colours are only used when the *TERM*[478] environment variable is set and either the terminal type can be found in *colour-terms*[356] or its name includes the string *color*.

On top of that the binary option *colour-pager*[283] defines whether these colour sequences are also generated when the output of a command needs to go through the *PAGER*[474] (also see *crt*[359]) – this is not enabled by default because different pager programs need different command line switches or other configuration in order to support those colour sequences, please see the option for more details.

To forcefully disable all colour support, set *colour-disable*[282].

Colours can be configured through font attributes (*ft*= *bold*, *invers* and *underline*), foreground (*fg*=) and background (*bg*=) colours (*black*, *blue*, *green*, *red*, *brown*, *magenta*, *cyan* and *white*). Multiple specifications can be joined in a comma separated list, as in

```
set colour-msginfo="ft=bold,fg=magenta,bg=cyan"
```

Options to be set are *colour-msginfo*[354], *colour-partinfo*[355], *colour-from_*[352], *colour-header*[353] and *colour-uheader*[357], as well as *colour-user-headers*[358], which is a list of headers to be colourized via *colour-uheader*[357] instead of the default *colour-header*[353].

COMMANDS[5]

Each command is typed on a line by itself, and may take arguments following the command word. The command need not be typed in its entirety – the first command which matches the typed prefix is used. (The command **list**[127] prints a sorted list of available commands, and the command **help**[122] (or **?[73]**), when given an argument, will show a documentation string for the expansion, as in `?unc`; documentation strings are however [Option].)

For commands which take message lists as arguments, the next message forward that satisfies the command's requirements will be used shall no explicit message list have been passed. If there are no messages forward of the current message, the search proceeds backwards, and if there are no good messages at all, S-nail types "no applicable messages" and aborts the command. The arguments to commands can be quoted, using the following methods:

- An argument can be enclosed between paired double-quotes "argument" or single-quotes 'argument'; any white space, shell word expansion, or backslash characters (except as described next) within the quotes are treated literally as part of the argument. A double-quote will be treated literally within single-quotes and vice versa. Inside such a quoted string the actually used quote character can be used nonetheless by escaping it with a backslash '\', as in "y\"ou".
- An argument that is not enclosed in quotes, as above, can usually still contain space characters if those spaces are backslash-escaped.

- A backslash outside of the enclosing quotes is discarded and the following character is treated literally as part of the argument.
- An unquoted backslash at the end of a command line is discarded and the next line continues the command.

Filenames, where expected, are subsequently subjected to the following transformations, in sequence:

- If the filename begins with an unquoted plus sign, and the *folder*[367] variable is defined, the plus sign will be replaced by the value of the *folder*[367] variable followed by a slash. If the *folder*[367] variable is unset or is set to null, the filename will be unchanged.
- Shell word expansions are applied to the filename. **Compatibility note:** on the long run support for complete shell word expansion will be replaced by an internally implemented restricted expansion mechanism in order to circumvent possible security impacts through shell expansion. Expect that a growing number of program parts only support this restricted syntax:

Meta expansions are applied to the filename: leading tilde characters (`~/`) will be replaced by the expansion of HOME [462], and any occurrence of `$VARIABLE` (or `${VARIABLE}`) will be replaced by the expansion of the variable, if possible; S-nail internal as well as environmental (shell) variables can be accessed through this mechanism. In order to include a raw `'$'` character precede it with a backslash `'\'`, to include a backslash double it. If more than a single pathname results from this expansion and the command is expecting one file, an error results.

Note that in interactive display context, in order to allow simple value acceptance (typing "ENTER"), backslash quoting is performed automatically as necessary, e.g., a file `diet\ is \curd.txt` will be displayed as `diet\\ is \\curd.txt`.

The following commands are available:

- # The comment-command causes the entire line to be ignored. **Note:** this really is a normal command which' purpose is to discard its arguments, not a "comment-start" indicating special character, which means that, e.g., trailing comments on a line are not possible.
- ~ Interprets the remainder of the word as a macro name and passes it through to the **call**[80] command; e.g., `~mymacro` is a shorter synonym for `call mymacro`.
- Print out the preceding message. If given a numeric argument *n*, goes to the *n*'th previous message and prints it.
- = Show the current message number (the "dot").
- ? Prints a brief summary of commands. [Option] Given an argument a synopsis for the command in question is printed instead; commands can be abbreviated in general and this command can be used to see the full expansion of an abbreviation including the synopsis, try, e.g., `'?h'`, `?hel` and `?help` and see how the display changes.
- ! Executes the SHELL [476] (see `sh(1)`) command which follows.
- | A synonym for the **pipe**[147] command.
- account** (ac) Creates, selects or lists (an) account(s). An account is a group of commands and variable settings which together usually arrange the environment for the purpose of creating a system login. Without any argument a listing of all defined accounts and their content is printed. A specific account can be activated by giving solely its name, resulting in the system mailbox or inbox of that account to be activated as via an explicit use of the command **file**[105]. The two argument form is identical to defining a macro as via **define**[91]. For example:

```
account myisp {
    set folder=mail
    set record=+Sent
```

```

    set from="myname@myisp.example (My Name)"
    set smtp=smtp://mylogin@smtp.myisp.example
}

```

creates an account named `myisp` which can later be selected by specifying `account myisp`. The special account `null` (case-insensitive) always exists. `localopts[128]` can be used to localize account settings – different to normal macros the settings will be reverted once the account is switched off. Accounts can be deleted via `unaccount [192]`.

alias (a) With no arguments, prints out all currently-defined aliases. With one argument, prints out that alias. With more than one argument, creates a new alias or appends to an existing one. `unalias[193]` can be used to delete aliases.

alternates

(alt) Manage a list of alternate addresses / names of the active user, members of which will be removed from recipient lists when replying to messages (and the `metoo[307]` variable is not set). If arguments are given the set of alternate names is replaced by them, without arguments the current set is displayed.

answered

Takes a message list and marks each message as having been answered. This mark has no technical meaning in the mail system; it just causes messages to be marked in the header summary, and makes them specially addressable.

call Calls a macro that has been created via `define[91]`.

cd (ch) Change the working directory to `HOME[462]` or the given argument. Synonym for `chdir[83]`.

certsave

[Option] Only applicable to S/MIME signed messages. Takes a message list and a file name and saves the certificates contained within the message signatures to the named file in both human-readable and PEM format. The certificates can later be used to send encrypted messages to the respective message senders by setting `smime-encrypt-USER@HOST[419]` variables.

chdir (ch) Change the working directory to `HOME[462]` or the given argument. Synonym for `cd[81]`.

collapse

Only applicable to threaded mode. Takes a message list and makes all replies to these messages invisible in header summaries, unless they are in state new.

connect [Option] If operating in disconnected mode on an IMAP mailbox, switch to online mode and connect to the mail server while retaining the mailbox status. See the description of the `disconnected` variable for more information.

Copy (C) Copy messages to files whose names are derived from the author of the respective message and don't mark them as being saved; otherwise identical to `Save[167]`.

copy (c) Copy messages to the named file and don't mark them as being saved; otherwise identical to `save[168]`.

cwd Print the current working directory.

Decrypt [Option] For unencrypted messages this command is identical to `Copy[86]`; Encrypted messages are first decrypted, if possible, and then copied.

decrypt [Option] For unencrypted messages this command is identical to **copy**[87]; Encrypted messages are first decrypted, if possible, and then copied.

define Without arguments the current list of macros, including their content, is printed, but otherwise a macro is defined. A macro definition is a sequence of commands in the following form:

```
define name {
    command1
    command2
    ...
    commandN
}
```

A defined macro can be invoked explicitly by using the **call**[80] or **~**[70] commands, or implicitly by setting the *folder-hook*[368] or *folder-hook-FOLDER*[369] variables. Note that interpretation of **localopts**[128] depends on how (i.e., “as what”: normal macro, folder hook, account switch) the macro is invoked. Macros can be deleted via **undefine** [196].

delete (d) Marks the given message list as deleted. Deleted messages will neither be saved in **MBOX**[470] nor will they be available for most other commands.

discard (di) Identical to **ignore**[126]. Also see **retain** [166].

dp, dt Deletes the current message and prints the next message. If there is no next message, S-nail says “at EOF”.

draft Takes a message list and marks each given message as a draft. This mark has no technical meaning in the mail system; it just causes messages to be marked in the header summary, and makes them specially addressable.

echo (ec) Echoes its arguments, resolving special names as documented for the command **file**[105]. The escape sequences ‘\a’, ‘\b’, ‘\c’, ‘\f’, ‘\n’, ‘\r’, ‘\t’, ‘\v’, ‘\\’ and ‘\octal-num’ are interpreted just as they are by `printf(1)` (proper quoting provided).

edit (e) Point the text editor (as defined in **EDITOR**[461]) at each message from the given list in turn. Modified contents are discarded unless the *writeback kedited*[342] variable is set.

elif Part of the **if**[125]/**elif**[99]/**else**[100]/**endif**[101] conditional — if the condition of a preceding **if**[125] was false, check the following condition and execute the following block if it evaluates true.

else (el) Part of the **if**[125]/**elif**[99]/**else**[100]/**endif**[101] conditional — if none of the conditions of the preceding **if**[125] and **elif**[99] commands was true, the **else**[100] block is executed.

endif (en) Marks the end of an **if**[125]/**elif**[99]/**else**[100]/**endif**[101] conditional execution block.

exit (ex or x) Exit from S-nail without changing the active mailbox and skip any saving of messages in **MBOX**[470] as well as a possibly tracked command line editor history file.

features

Print the list of features that have been compiled into S-nail.

File (Fi) Like **file**[105], but open the mailbox readonly.

file (fi) The file command switches to a new mailbox. Without arguments it prints the complete state of the current mailbox. If an argument is given, it will write out changes (such as deletions) the user has made and open a new mailbox. Some special conventions are recognized for the *name* argument:

(number sign) means the previous file,
 % (percent sign) means the invoking user's system mailbox,
 %*user* means the system mailbox of *user* (and never the value of *folder*[367], regardless of its actual setting),
 & (ampersand) means the invoking user's MBOX[470] file and
 +*file* means a *file* in the *folder*[367] directory.
 %:*filespec* expands to the same value as *filespec*, but the file is handled as a system mailbox by, e.g., the **mbox**[132] and **save**[168] commands, meaning that messages that have been read in the current session will be moved to the MBOX[470] mailbox instead of simply being flagged as read.

If the name matches one of the strings defined with the command **shortcut**[176], it is replaced by its long form and expanded. If the name ends with .gz, .bz2 or .xz it is treated as being compressed with **gzip**(1), **bzip2**(1) or **xz**(1), respectively, and transparently handled through an intermediate (un)compression step (using a temporary file) with the according facility, sufficient support provided. Likewise, if the named file doesn't exist, but a file with one of the mentioned compression extensions does, then the name is automatically expanded and the compressed file is used. Also see *mbox-rfc4155*[305] for fine-tuning the handling of MBOX files.

If *name* refers to a directory with the subdirectories tmp, new and cur, then it is treated as a folder in "Maildir" format. A name of the form

```
[v15-compat] protocol://[user[:password]@]host[:port][/path]
[no v15-compat] protocol://[user@]host[:port][/path]
```

is taken as an Internet mailbox specification. The [Option]ally supported protocols are *pop3* (POP3) and *pop3s* (POP3 with SSL/TLS encrypted transport). The *[/path]* part is valid only for IMAP; there it defaults to *INBOX*. Also see the section **On URL syntax and credential lookup**[26].

[no v15-compat] If *user* contains special characters, in particular '/' or '%', they must be escaped in URL notation – the command **urlencode**[218] can be used to show the necessary conversion.

flag Takes a message list and marks the messages as **flag**[106]ged for urgent/special attention. This mark has no technical meaning in the mail system; it just causes messages to be highlighted in the header summary, and makes them specially addressable.

folder (fold) The same as **file**[105].

folders With no arguments, list the names of the folders in the folder directory. With an existing folder as an argument, lists the names of folders below the named folder.

Followup

(F) Similar to **Respond**[162], but saves the message in a file named after the local part of the first recipient's address.

followup

(fo) Similar to **respond**[163], but saves the message in a file named after the local part of the first recipient's address.

followupall

Similar to **followup**[110], but responds to all recipients regardless of the *flipr*[290] variable.

followupsender

Similar to **Followup**[109], but responds to the sender only regardless of the *flipr*[290] variable.

Forward Alias for **Fwd**[116].

forward Alias for **fwd**[117].

from (f) Takes a list of messages and prints their message headers.

Fwd Similar to **fwd**[117], but saves the message in a file named after the local part of the recipient's address.

fwd Takes a message and the address of a recipient and forwards the message to him. The text of the original message is included in the new one, with the value of the *fwdheading*[372] variable printed before. The **fwdignore**[118] and **fwdretain**[119] commands specify which header fields are included in the new message. Only the first part of a multipart message is included unless the *forward-as-attachment*[292] option is set.

fwdignore

Specifies which header fields are to be ignored with the command **fwd**[117]. This command has no effect when the *forward-as-attachment*[292] option is set.

fwdretain

Specifies which header fields are to be retained with the command **fwd**[117]. **fwdretain**[119] overrides **fwdignore**[118]. This command has no effect when the *forward-as-attachment*[292] option is set.

ghost Define or list command aliases, so-called ghosts. Without arguments a list of all currently known aliases is printed. With one argument the expansion of the given alias is shown. With two or more arguments a command alias is defined or updated: the first argument is the name under which the remaining command line should be accessible, the content of which can be just about anything. A ghost can be used everywhere a normal command can be used, but always takes precedence; any arguments that are given to the command alias are joined onto the alias content, and the resulting string forms the command line that is, in effect, executed. Also see **unghost**[202].

```
? gh xx
'ghost': no such alias: "xx"
? gh xx echo hello,
? gh xx
ghost xx "echo hello,"
? xx
hello,
? xx world
hello, world
```

headers (h) Show the current group of headers, the size of which depends on the variable *screen*[405]. If a message-specification is given the group of headers containing the first message therein is shown and the message at the top of the screen becomes the new “dot”.

help (hel) A synonym for **?**[73].

history [Option] Either *show* or *clear* the list of history entries; a decimal *NUMBER* argument selects and shows the respective history entry – press “ENTER” to accept it, and the history entry will become the new history top. The default mode if no arguments are given is *show*.

hold (ho, also **preserve**[148]) Takes a message list and marks each message therein to be saved in the user's system mailbox instead of in *MBOX*[470]. Does not override the **delete**[92] command. S-nail deviates from the POSIX standard with this command, because a **next**[142] command issued after **hold**[124] will display the following message, not the current one.

if (i) Part of the nestable **if[125]/elif[99]/else[100]/endif[101]** conditional execution construct — if the given condition is true then the encapsulated block is executed. POSIX only supports the conditions **[Rr]eceive**, **[Ss]end** and **[Tt]erm** (execute if standard input is a tty), all remaining conditions are non-portable extensions; note that falsely specified conditions cause the execution of the entire conditional construct until the (matching) closing **endif[101]** command to be suppressed. The syntax of the nestable **if[125]** conditional execution construct requires that each condition and syntax element is surrounded by whitespace.

```
if receive
  commands ...
else
  commands ...
endif
```

Another condition can be any boolean value (see the section **Value options[31]** for booleans) to mark an enwrapped block as “never execute” or “always execute”. It is possible to check a variable for existence or compare its expansion against a user given value or another variable via the ‘\$’ (“variable next”) conditional trigger character. The available comparison operators are ‘<’ (less than), ‘<=’ (less than or equal to), ‘==’ (equal), ‘!=’ (not equal), ‘>=’ (greater than or equal to), ‘>’ (greater than), ‘=@’ (is substring of) and ‘!@’ (is not substring of). The values of the left and right hand side are treated as strings and are compared 8-bit byte-wise, ignoring case according to the rules of the US-ASCII encoding (therefore, dependend on the active locale, possibly producing false results for strings in the locale encoding). Except for the substring checks the comparison will instead be performed arithmetically if both, the user given value as well as the variable content, can be parsed as numbers (integers). An unset variable is treated as the empty string.

When the [Option]al regular expression support is available, the additional test cases ‘=~’ and ‘!~’ can be used. They treat the right hand side as a regular expression that is matched case-insensitively and according to the active **LC_CTYPE[466]** locale, meaning that strings in the locale encoding should be matched correctly.

Conditions can be joined via AND-OR lists (where the AND operator is ‘&&’ and the OR operator is ‘|’), which have equal precedence and will be evaluated with left associativity, thus using the same syntax that is known for the **sh(1)**. It is also possible to form groups of conditions and lists by enclosing them in pairs of brackets [...], which may be interlocked within each other, and also be joined via AND-OR lists.

The results of individual conditions and entire groups may be modified via unary operators: the unary operator ‘!’ will reverse the result.

```
if $debug
  echo *debug* is set
endif
if $ttycharset == "UTF-8"
  echo *ttycharset* is set to UTF-8, case-insensitively
endif
set t1=one t2=one
if $t1 == $t2
  echo These two variables are equal
endif
if $version-major >= 15
  echo Running a new version..
  if $features =@ "regex"
    if $TERM =~ "^xterm.*"
```

```

        echo ..in an X terminal
    endif
endif
if [ [ true ] && [ [ $debug ] || [ $verbose ] ] ]
    echo Noisy, noisy
endif
if true && $debug || $verbose
    echo Left associativity, as is known from the shell
endif
if ! ! true && ! [ ! $debug && ! $verbose ]
    echo Unary operator support
endif
endif
endif

```

ignore Without arguments the list of ignored header fields is printed, otherwise the given list of header fields is added to the ignore list: Header fields in the ignore list are not printed on the terminal when a message is printed. To print a message in its entirety, use the commands **Type**[190] or **Print**[149]. Also **seediscard** [93] and **retain**[166].

list Prints the names of all available commands, alphabetically sorted.

localopts

This command can be used to localize changes to variables, meaning that their state will be reverted to the former one once the covered scope is left. It can only be used inside of macro definition blocks introduced by **account**[76] or **define**[91], and is interpreted as a boolean (see **Value options**[31]); the “covered scope” of an account is left once it is switched off again.

```

define temporary_settings {
    set global_option1
    localopts on
    set local_option1
    set local_option2
    localopts off
    set global_option2
}

```

Note that this setting “stacks up”: i.e., if `macro1` enables change localization and calls `macro2`, which explicitly resets localization, then any value changes within `macro2` will still be reverted by `macro1`!

lreply Reply to messages that come in via known (**mlist**[134]) or subscribed (**mlsubscribe**[135]) mailing lists, or pretend to do so (see **Mailing lists**[21]): on top of the usual **reply**[157] functionality this will actively resort and even remove message recipients in order to generate a message that is supposed to be send to a mailing list. For example it will also implicitly generate a `Mail-Followup-To:` header if that seems useful, regardless of the setting of the variable *followup-to*[291].

Mail Similar to **mail**[131], but saves the message in a file named after the local part of the first recipient’s address.

mail (m) Takes a (list of) recipient address(es) as (an) argument(s), or asks on standard input if none were given; then collects the remaining mail content and sends it out.

mbox (mb) The given message list is to be sent to **MBOX**[470] when S-nail is quit; this is the default action unless the *hold*[297] option is set. S-nail deviates from the POSIX standard with this command, as a **next**[142] command issued after **mbox**[132] will display the following mes-

sage, not the current one.

mimetype

Without any arguments the content of the MIME type cache will be displayed. Otherwise each argument defines a complete MIME type specification of a type that shall be added (prepended) to the cache. In any event MIME type sources are loaded first as necessary – *mimetypes-load-control*[380] can be used to fine-tune which sources are actually loaded. Refer to the section on **The mime.types files**[32] for more on MIME type specifications and this topic in general. MIME type unregistration and cache resets can be triggered with **unmimetype**[204].

mlist

Without arguments the list of all currently defined mailing lists (and their attributes, if any) is printed. Otherwise all given arguments (which need not be quoted except for whitespace) will be added and henceforth be recognized as mailing lists. Mailing lists may be removed via the command **unmlist**[205].

If the [Option]al regular expression support is available then mailing lists may also be specified as regular expressions (see *re_format*(7) for more on those).

mlsubscribe

Without arguments the list of all currently defined mailing lists which have a subscription attribute is printed. Otherwise this attribute will be set for all given mailing lists, newly creating them as necessary (as via **mlist**[134]). Subscription attributes may be removed via the command **unmlsubscribe**[206]. Also see *followup-to* [291].

move

Acts like **copy**[87] but marks the messages for deletion if they were transferred successfully.

more

Like **print**[150] or **type**[191], but prints a form-feed ('\f') in between messages.

More

Like **more**[137], but also prints ignored header fields and all MIME parts.

Move

Similar to **move**[136], but moves the messages to a file named after the local part of the sender address of the first message.

netrc

[Option] Either (*show* or *clear*) the current *.netrc* cache, loading the file first as necessary in the former case. Note that S-nail will try to read the file only once, use **netrc**[140] *clear* to unlock the next attempt. See *netrc-lookup*[309] and the section **On URL syntax and credential lookup**[26]; the section **The .netrc file**[33] documents the file format in detail.

newmail

Checks for new mail in the current folder without committing any changes before. If new mail is present, a message is printed. If the *header*[294] variable is set, the headers of each new message are also printed.

next

(n) (like “+” or “ENTER”) Goes to the next message in sequence and types it. With an argument list, types the next matching message.

New

Same as **Unread**[207].

new

Same as **unread**[208].

noop

If the current folder is accessed via a network connection, a “NOOP” command is sent, otherwise no operation is performed.

Pipe

Like **pipe**[147] but also pipes ignored header fields and all parts of MIME multipart/alternative messages.

pipe

(pi) Takes a message list and a shell command and pipes the messages through the command. Without an argument the current message is piped through the command given by the *cmd*[351] variable. If the *pa ge*[311] variable is set, every message is followed by a formfeed character.

preserve

(pre) A synonym for **hold**[124].

Print (P) Like **print**[150] but also prints out ignored header fields and all parts of MIME multipart/alternative messages. See also **print**[150], **ignore**[126] and **retain**[166].

print (p) Takes a message list and types out each message on the user's terminal. For MIME multipart messages, all parts with a content type of `text` or `message` are shown, the other are hidden except for their headers. Messages are decrypted and converted to the terminal character set if necessary.

quit (q) Terminates the session, saving all undeleted, unsaved messages in the current MBOX[470], preserving all messages marked with **hold**[124] or **preserve**[148] or never referenced in the system mailbox, and removing all other messages from the system mailbox. If new mail has arrived during the session, the message "You have new mail" will be shown. If given while editing a mailbox file with the command line flag **-f** [47], then the edit file is rewritten. A return to the shell is effected, unless the rewrite of edit file fails, in which case the user can escape with the exit command.

redirect

Same as **resend**[161].

Redirect

Same as **Resend**[160].

remove Removes the named folders. The user is asked for confirmation in interactive mode.

rename Takes the name of an existing folder and the name for the new folder and renames the first to the second one. Both folders must be of the same type.

Reply (R) Reply to originator. Does not reply to other recipients of the original message. *flpr*[290] will exchange this command with **reply**[157].

reply (r) Take a message and group-responds to it by addressing the sender and all recipients. *followup-to*[291], *followup-to-honour*[370], *reply-to-honour*[404] as well as *recipients-in-cc*[321] influence response behaviour. The command **Lreply** [129] offers special support for replying to mailing lists. If *flpr*[290] is set the commands **Reply**[156] and **reply**[157] are exchanged.

replyall

Similar to **reply**[157], but initiates a group-reply regardless of the value of *flpr*[290].

replysender

Similar to **Reply**[156], but responds to the sender only regardless of the value of *flpr*[290].

Resend Like **resend**[161], but does not add any header lines. This is not a way to hide the sender's identity, but useful for sending a message again to the same recipients.

resend Takes a list of messages and a user name and sends each message to the named user. `Resent-From:` and related header fields are prepended to the new copy of the message.

Respond Same as **Reply**[156].

respond Same as **reply**[157].

respondall

Same as **replyall**[158].

respondsender

Same as **replysender**[159].

retain (ret) Without arguments the list of retained header fields is printed, otherwise the given list of header fields is added to the retain list: Header fields in the retain list are shown on the terminal when a message is printed, all other header fields are suppressed. To print a message in its entirety, use the commands **Type**[190] or **Print**[149]. Also **seediscard** [93] and **ignore**[126]; **retain**[166] takes precedence over the mentioned.

Save (S) Similar to **save**, but saves the messages in a file named after the local part of the sender of the first message instead of taking a filename argument.

save (s) Takes a message list and a filename and appends each message in turn to the end of the file. If no filename is given, the MBOX[470] file is used. The filename in quotes, followed by the generated character count is echoed on the user's terminal. If editing a system mailbox the messages are marked for deletion. Compressed files are handled as described for the **-f**[47] command line option above.

savediscard

Same as **saveignore**[170].

saveignore

Is to **save**[168] what **ignore**[126] is to **print**[150] and **type**[191]. Header fields thus marked are filtered out when saving a message by **save**[168] or when automatically saving to MBOX[470]. This command should only be applied to header fields that do not contain information needed to decode the message, as MIME content fields do.

saveretain

Is to **save**[168] what **retain**[166] is to **print**[150] and **type**[191]. Header fields thus marked are the only ones saved with a message when saving by **save**[168] or when automatically saving to MBOX[470]. **saveretain**[171] overrides **saveignore**[170]. The use of this command is strongly discouraged since it may strip header fields that are needed to decode the message correctly.

seen Takes a message list and marks all messages as having been read.

set (se) With no arguments, prints all variable values. Otherwise, sets an option. Arguments are of the form `option=value` (no space before or after '='), or plain `option` if there is no value. Quotation marks may be placed around any part of the assignment statement to quote blanks or tabs, e.g.,

```
set indentprefix="->"
```

If an argument begins with 'no', as in `set nosave`, the effect is the same as invoking the **unset**[212] command with the remaining part of the variable (`unset save`).

setenv Identical to **set**[173] except that the options are also exported into the program environment; since this task requires native host support the command will always report error if that is not available (but still act like **set**[173] in this case). This operation is a no-op unless all resource files have been loaded. Also see **unsetenv**[213].

shell (sh) Invokes an interactive version of the shell.

shortcut

Without arguments the list of all currently defined shortcuts is printed. Otherwise all given arguments (which need not be quoted except for whitespace) are treated as pairs of shortcuts and their expansions, creating new or changing already existing shortcuts, as necessary. Shortcuts may be removed via the command **unshortcut**[214]. The expansion strings should be in the syntax

that has been described for the **file**[105] command.

- show** Like **print**[150], but performs neither MIME decoding nor decryption, so that the raw message text is shown.
- size** (si) Print the size in characters of each message of the given message-list.
- sort** Create a sorted representation of the current folder, and change the **next**[142] command and the addressing modes such that they refer to messages in the sorted order. Message numbers are the same as in regular mode. If the *header*[294] variable is set, a header summary in the new order is also printed. Possible sorting criteria are:
- | | |
|---------|---|
| date | Sort the messages by their <code>Date</code> : field, that is by the time they were sent. |
| from | Sort messages by the value of their <code>From</code> : field, that is by the address of the sender. If the <i>showname</i> [330] variable is set, the sender's real name (if any) is used. |
| size | Sort the messages by their size. |
| spam | [Option] Sort the message by their spam score, as has been classified by spamrate [184]. |
| status | Sort the messages by their message status. |
| subject | Sort the messages by their subject. |
| thread | Create a threaded display. |
| to | Sort messages by the value of their <code>To</code> : field, that is by the address of the recipient. If the <i>showname</i> [330] variable is set, the recipient's real name (if any) is used. |

If no argument is given, the current sorting criterion is printed.

source (so) The source command reads commands from a file.

spamclear

[Option] Takes a list of messages and clears their `is-spam` flag.

spamforget

[Option] Takes a list of messages and causes the *spam-interface*[425] to forget it has ever used them to train its Bayesian filter. Unless otherwise noted the `is-spam` flag of the message is inspected to chose wether a message shall be forgotten to be “ham” or “spam”.

spamham [Option] Takes a list of messages and informs the Bayesian filter of the *spam-interface*[425] that they are “ham”. This also clears the `is-spam` flag of the messages in question.

spamrate

[Option] Takes a list of messages and rates them using the configured *spam-interface*[425], without modifying the messages, but setting their `is-spam` flag as appropriate; because the spam rating headers are lost the rate will be forgotten once the mailbox is left. Refer to the manual section **Handling spam**[37] for the complete picture of spam handling in S-nail.

spamset [Option] Takes a list of messages and sets their `is-spam` flag.

spamspam

[Option] Takes a list of messages and informs the Bayesian filter of the *spam-interface*[425] that they are “spam”. This also sets the `is-spam` flag of the messages in question.

thread [Obsolete] The same as `sort thread` (consider using a `ghost` as necessary). Create a threaded representation of the current folder, i.e. indent messages that are replies to other messages in the header display and change the **next**[142] command and the addressing modes such that they refer to messages in the threaded order. Message numbers are the same as in unthreaded mode. If the *header*[294] variable is set, a header summary in threaded order is also

printed.

top (to) Takes a message list and prints the top few lines of each. The number of lines printed is controlled by the variable *toplines*[452] and defaults to five.

touch (tou) Takes a message list and marks the messages for saving in MBOX[470]. S-nail deviates from the POSIX standard with this command, as a following **next**[142] command will display the following message instead of the current one.

Type (T) Identical to the **Print**[149] command.

type (t) A synonym for **print**[150].

unaccount

Delete all given accounts. An error message is printed if a given account is not defined. The special name '*' will discard all existing accounts.

unalias (una) Takes a list of names defined by alias commands and discards the remembered groups of users. The special name '*' will discard all existing aliases.

unanswered

Takes a message list and marks each message as not having been answered.

uncollapse

Only applicable to threaded mode. Takes a message list and makes the message and all replies to it visible in header summaries again. When a message becomes the current message, it is automatically made visible. Also when a message with collapsed replies is printed, all of these are automatically uncollapsed.

undefine

Undefine all given macros. An error message is printed if a given macro is not defined. The special name '*' will discard all existing macros.

undeleter

(u) Takes a message list and marks each message as not being deleted.

undraft Takes a message list and **undraft**[96]s each message.

unflag Takes a message list and marks each message as not being **flag**[106]ged.

unfwdignore

Removes the header field names from the list of ignored fields for the **forward**[114] command. The special name '*' will remove all fields.

unfwdretain

Removes the header field names from the list of retained fields for the **forward**[114] command. The special name '*' will remove all fields.

unghost Remove all the given command **ghost**[120]s. The special name '*' will remove all ghosts.

unignore

Removes the header field names from the list of ignored fields. The special name '*' will remove all fields.

unmimetype

Delete all given MIME types, e.g., `unmimetype text/plain` will remove all registered specifications for the MIME type `text/plain`. The special name '*' will discard all existing MIME types, just as will `reset`, but which also reenables cache initialization via *mimetypes-load-control*[380].

unmlist Forget about all the given mailing lists. The special name ‘*’ will remove all lists. Also see **mlist**[134].

unmlsubscribe

Remove the subscription attribute from all given mailing lists. The special name ‘*’ will clear the attribute from all lists which have it set. Also see **mlsubscribe**[135].

Unread Same as **unread**[208].

unread Takes a message list and marks each message as not having been read.

unretain

Removes the header field names from the list of retained fields. The special name ‘*’ will remove all fields.

unsaveignore

Removes the header field names from the list of ignored fields for saving. The special name ‘*’ will remove all fields.

unsaveretain

Removes the header field names from the list of retained fields for saving. The special name ‘*’ will remove all fields.

unset (uns) Takes a list of option names and discards their remembered values; the inverse of **set**[173].

unsetenv

Identical to **unset**[212] except that the options are also removed from the program environment; since this task requires native host support the command will always report error if that is not available (but still act like **unset**[212]). This operation is a no-op unless all resource files have been loaded. Also see **setenv**[174].

unshortcut

Deletes the shortcut names given as arguments. The special name ‘*’ will remove all shortcuts.

unsort Disable sorted or threaded mode (see the **sort**[179] and **thread**[187] commands), return to normal message order and, if the *header*[294] variable is set, print a header summary.

unthread

[Obsolete] Same as **unsort**[215].

urldecode

Decode the given URL-encoded string arguments and show the results.

urlencode

URL-encode the given arguments and show the results.

varedit Edit the values of or create the given variable(s) in the **EDITOR**[461]. Binary variables cannot be edited.

varshow Show informations about all the given options. S-nail knows about a finite set of known builtin variables that are subdivided further in binary and value variants; they may have special properties, like “read-only” (setting may not be changed) and “virtual”, meaning that the value is generated on-the-fly as necessary. Beside those known variables an infinite number of unknown, so-called “assembled” variables, which are expected to be able to store values, may exist.

```
? set foo=bar nobar
? varshow sendwait version-major foo bar
"sendwait": (73) binary: set=1 (ENVIRON=0)
"version-major": (192) value, read-only, virtual:\
```

```

    set=1 (ENVIRON=0) value<14>
"foo": (assembled) set=1 (ENVIRON=0) value<bar>
"bar": (assembled) set=0 (ENVIRON=0) value<NULL>

```

- verify** [Option] Takes a message list and verifies each message. If a message is not a S/MIME signed message, verification will fail for it. The verification process checks if the message was signed using a valid certificate, if the message sender's email address matches one of those contained within the certificate, and if the message content has been altered.
- visual** (v) Takes a message list and invokes the display editor on each message. Modified contents are discarded unless the *writebackedited*[342] variable is set.
- write** (w) For conventional messages the body without all headers is written. The output is decrypted and converted to its native format as necessary. If the output file exists, the text is appended. If a message is in MIME multipart format its first part is written to the specified file as for conventional messages, and the user is asked for a filename to save each other part. For convenience saving of each part may be skipped by giving an empty value; the same result can also be achieved by writing it to */dev/null*. For the second and subsequent parts a leading '|' character causes the part to be piped to the remainder of the user input interpreted as a shell command; otherwise the user input is expanded as usually for folders, e.g., tilde expansion is performed. In non-interactive mode, only the parts of the multipart message that have a filename given in the part header are written, the others are discarded. The original message is never marked for deletion in the originating mail folder. For attachments, the contents of the destination file are overwritten if the file previously existed. No special handling of compressed files is performed.
- xit** (x) A synonym for **exit**[102].
- z** S-nail presents message headers in windowfuls as described under the **headers**[121] command. This command scrolls to the next window of messages. If an argument is given, it specifies the window to use. A number prefixed by '+' or '-' indicates that the window is calculated in relation to the current position. A number without a prefix specifies an absolute window number, and a '\$' lets S-nail scroll to the last window of messages.
- Z** Similar to **z**[225], but scrolls to the next or previous window that contains at least one new or **flag**[106]ged message.

TILDE ESCAPES[6]

Here is a summary of the tilde escapes, which are used to perform special functions when composing messages. Tilde escapes are only recognized at the beginning of lines. The name "tilde escape" is somewhat of a misnomer since the actual escape character can be changed by adjusting the option *escape*[363].

- ~~** *string* Insert the string of text in the message prefaced by a single '~'. (If the escape character has been changed, that character must be doubled in order to send it at the beginning of a line.)
- ~!** *command* Execute the indicated shell *command*, then return to the message.
- ~.** Same effect as typing the end-of-file character.
- ~:** *S-nail-command* or **~_** *S-nail-command*
Execute the given S-nail command. Not all commands, however, are allowed.
- ~?** Write a summary of command escapes.
- ~<** *filename* Identical to **~r**[251].
- ~<!** *command* *command* is executed using the shell. Its standard output is inserted into the message.

`~@ [filename...]`

With no arguments, edit the attachment list interactively. If an attachment's file name is left empty, that attachment is deleted from the list. When the end of the attachment list is reached, S-nail will ask for further attachments until an empty name is given. If a given file name solely consists of the number sign '#' followed by a valid message number of the currently active mailbox, then the given message is attached as a MIME message/rfc822 and the rest of this section does not apply.

If character set conversion has been compiled into S-nail, then this mode gives the user the option to specify input and output character sets, unless the file extension indicates binary content, in which case S-nail asks whether this step shall be skipped for the attachment in question. If not skipped, then the charset that succeeds to represent the attachment data will be used in the `charset=` MIME parameter of the mail message:

- If input and output character sets are specified, then the conversion is performed on the fly. The user will be asked repeatedly until the desired conversion succeeds.
- If only an output character set is specified, then the input is assumed to be in the `tycharset[453]` charset and will be converted to the given output charset on the fly. The user will be asked repeatedly until the desired conversion succeeds.
- If no character sets are specified at all then the algorithm that is documented in the section **Character sets[23]** is applied, but directly and on the fly. The user will be asked repeatedly until the desired conversion succeeds.
- Finally, if an input-, but no output character set is specified, then no conversion is ever performed, but the `charset=` MIME parameter value will still be set to the user input.
- The character set selection loop can be left by typing `control-C`, i.e., causing an interrupt. Note that before S-nail version 15.0 this terminates the entire current attachment selection, not only the character set selection.

Without character set conversion support, S-nail will ask for the input character set only, and it'll set the `charset=` MIME parameter value to the given input, if any; if no user input is seen then the `tycharset[453]` character set will be used for the parameter value instead. Note that the file extension check isn't performed in this mode, since no conversion will take place anyway.

Note that in non-interactive mode, for reproducibilities sake, there will always be two questions for each attachment, regardless of whether character set conversion is available and what the file extension is. The first asks for the filename, and the second asks for the input character set to be passed through to the corresponding MIME parameter; no conversion will be tried if there is input to the latter question, otherwise the usual conversion algorithm, as above, is applied. For message attachments, the answer to the second question is completely ignored.

If (instead) `filename` arguments are specified for the `~@` command they are treated as a comma separated list of files, which are all expanded and appended to the end of the attachment list. (Filenames with commas, or with leading or trailing whitespace can only be added via the command line or the first method. Message attachments can only be added via the first method; filenames which clash with message numbers can only be added via the command line or the second method.) In this mode the (text) attachments are assumed to be in `tycharset[453]` encoding, and will be evaluated as documented in the section **Character sets[23]**.

`~A`

Inserts the string contained in the `Sign[411]` variable (same as `~i[245]` Sign). The escape sequences tabulator '\t' and newline '\n' are understood.

- ~a** Inserts the string contained in the *sign*[412] variable (same as **~i**[245] sign). The escape sequences tabulator ‘\t’ and newline ‘\n’ are understood.
- ~b name . . .** Add the given names to the list of blind carbon copy recipients.
- ~c name . . .** Add the given names to the list of carbon copy recipients.
- ~d** Read the file specified by the *DEAD*[460] variable into the message.
- ~e** Invoke the text editor on the message collected so far. After the editing session is finished, the user may continue appending text to the message.
- ~F messages** Read the named messages into the message being sent, including all message headers and MIME parts. If no messages are specified, read in the current message.
- ~f messages** Read the named messages into the message being sent. If no messages are specified, read in the current message. **ignore**[126] and **retain**[166] lists are used to modify the message headers. For MIME multipart messages, only the first printable part is included.
- ~H** Edit the message header fields *From:*, *Reply-To:*, *Sender:* and *Organization:* by typing each one in turn and allowing the user to edit the field. The default values for these fields originate from the *fr om*[371], *replyto*[403], *sender*[407] and *ORGANIZATION*[388] variables.
- ~h** Edit the message header fields *To:*, *Cc:*, *Bcc:* and *Subject:* by typing each one in turn and allowing the user to edit the field.
- ~i variable** Insert the value of the specified variable into the message, adding a newline character at the end. The message remains unaltered if the variable is unset or empty. The escape sequences tabulator ‘\t’ and newline ‘\n’ are understood.
- ~M messages** Read the named messages into the message being sent, indented by *indentprefix*[376]. If no messages are specified, read the current message.
- ~m messages** Read the named messages into the message being sent, indented by *indentprefix*[376]. If no messages are specified, read the current message. **ignore**[126] and **retain**[166] lists are used to modify the message headers. For MIME multipart messages, only the first printable part is included.
- ~p** Print out the message collected so far, prefaced by the message header fields and followed by the attachment list, if any.
- ~q** Abort the message being sent, copying it to the file specified by the *DEAD*[460] variable if *save*[325] is set.
- ~R filename** Read the named file into the message, indented by *indentprefix*[376].
- ~r filename** Read the named file into the message.
- ~s string** Cause the named string to become the current subject field.
- ~t name . . .** Add the given name(s) to the direct recipient list.
- ~U messages** Read in the given / current message(s) excluding all headers, indented by *indentprefix*[376].

- `~u messages` Read in the given / current message(s), excluding all headers.
- `~v` Invoke an alternate editor (defined by the `VISUAL` [481] option) on the message collected so far. Usually, the alternate editor will be a screen editor. After the editor is quit, the user may resume appending text to the end of the message.
- `~w filename` Write the message onto the named file. If the file exists, the message is appended to it.
- `~x` Same as `~q` [249], except that the message is not saved at all.
- `~| command` Pipe the message through the specified filter command. If the command gives no output or terminates abnormally, retain the original text of the message. E.g., the command `fmt(1)` is often used as a rejustifying filter.

VARIABLE OPTIONS[7]

Variables are controlled via `set` [173] and `unset` [212] commands; in general using `unset` [212] can also be accomplished by prefixing a variable name with the string “no” and calling `set` [173], e.g., `unset crt` will have the same effect as `set nocrt`. Creation or editing of variables in an editor can also be achieved with `varedit` [219]. `varshow` [220] will give more insight on the given variable(s), whereas `set` [173] will print a listing of all variables when called without arguments. Options are also implicitly inherited from the program `ENVIRONMENT` [8] and can be set explicitly via the command line option `-s` [57].

Different kind of options exist: binary options, which can only be in one of the two states “set” and “unset”, as well as value options which have an assigned string value, for which proper quoting may be important upon assignment time.

Initial Settings[29]

The standard POSIX 2008/Cor 1-2013 mandates the following initial variable settings: `noallnet` [261], `noappend` [262], `asksub` [264], `noaskbcc` [268], `noautoprint` [271], `nobang` [273], `nocmd` [351], `nocrt` [359], `nodebug` [284], `nodot` [286], `escape` [363] set to “~”, `noflpr` [290], `nofolder` [367], `header` [294], `nohold` [297], `noignore` [299], `noignoreeof` [300], `nokeep` [301], `nokeepsave` [302], `nometoo` [307], `nooutfolder` [310], `nopage` [311], `prompt` [398] set to “?” (note that S-nail deviates from the standard by using “&”, but the “&” special prompt escape results in “?” being printed unless `bsdcompat` [276] is set), `noquiet` [319], `norecord` [401], `save` [325], `nosendwait` [328], `noshowto` [331], `noSign` [411], `nosign` [412], `toplines` [452] set to “5”.

Notes: S-nail doesn’t support the `noonehop` variable – use command line options or `sendmail-arguments` [409] to pass options through to a MTA.

Binary options[30]

add-file-recipients

When file or pipe recipients have been specified, mention them in the corresponding address fields of the message instead of silently stripping them from their recipient list. By default such addressees are not mentioned.

allnet Causes only the local part to be evaluated when comparing addresses.

append Causes messages saved in `MBOX` [470] to be appended to the end rather than prepended. This should always be set.

ask or *asksub*

Causes S-nail to prompt for the subject of each message sent. If the user responds with simply a newline, no subject field will be sent.

- askatend* Causes the prompts for Cc : and Bcc : lists to appear after the message has been edited.
- askattach* If set, S-nail asks for files to attach at the end of each message. An empty line finalizes the list.
- askcc* Causes the user to be prompted for additional carbon copy recipients (at the end of each message if *askatend*[265] or *bsdcompat*[276] are set). An empty line finalizes the list.
- askbcc* Causes the user to be prompted for additional blind carbon copy recipients (at the end of each message if *askatend*[265] or *bsdcompat*[276] are set). An empty line finalizes the list.
- asksign* [Option] Causes the user to be prompted if the message is to be signed at the end of each message. The *mime-sign* [334] variable is ignored when this variable is set.
- autocollapse*
Causes threads to be collapsed automatically when threaded mode is entered (see the **collapse**[84] command).
- autoprint* Causes the delete command to behave like `dp -`; thus, after deleting a message the next one will be typed automatically.
- autothread*
[Obsolete] Causes threaded mode (see the **thread**[187] command) to be entered automatically when a folder is opened. The same as `autosort=thread`.
- bang* Enables the substitution of ‘!’ by the contents of the last command line in shell escapes.
- batch-exit-on-error*
If the batch mode has been enabled via the `-#[65]` command line option, then this variable will be consulted whenever S-nail completes one operation (returns to the command prompt); if it is set then S-nail will terminate if the last operation generated an error.
- bsdannounce*
Causes automatic display of a header summary after executing a **file**[105] command.
- bsdcompat*
Sets some cosmetical features to traditional BSD style; has the same affect as setting *askatend*[265] and all other variables prefixed with “bsd”; it also changes the meaning of the S-nail specific ‘\&’ *prompt*[398] escape sequence.
- bsdflags* Changes the letters printed in the first column of a header summary to traditional BSD style.
- bsdheadline*
Changes the display of columns in a header summary to traditional BSD style.
- bsdmsgs* Changes some informational messages to traditional BSD style.
- bsdorder* Causes the Subject : field to appear immediately after the To : field in message headers and with the **~h**[244] **TILDE ESCAPES**[6].
- bsdset* Changes the output format of the **set**[173] command to traditional BSD style.
- colour-disable*
[Option] Forcefully disable usage of colours. Also see the section **Coloured message display**[28].
- colour-pager*
[Option] Whether colour shall be used for output that is paged through **PAGER**[474]. Note that pagers may need special flags, e.g., `less(1)` requires the option `-R` and `lv(1)` the option `-c` in order to support colours; therefore S-nail will inspect the variable **PAGER**[474] – if that starts with the string “less” a non-existing environment variable *LESS* will be set to `FRSX!`, likewise for “lv” *LV* will optionally be set to “-c”. Also see the section **Coloured message display**[28]

for more on this.

debug Prints debugging messages and disables the actual delivery of messages.

disposition-notification-send

[Option] Emit a `Disposition-Notification-To:` header (RFC 3798) with the message. This requires a set *from*[371] variable.

dot

When dot is set, a dot ‘.’ on a line by itself during message input from a terminal shall be treated as end-of-message (in addition to the normal end-of-file condition). If *ignoreeof*[300] is set *nodot*[286] is ignored and using a dot is the only method to terminate input mode.

editalong

If this variable is set then the editor is started automatically when a message is composed in interactive mode, as if the `~e`[240] **TILDE ESCAPES**[6] had been specified. The *editheaders*[288] variable is implied for this automatically spawned editor session.

editheaders

When a message is edited while being composed, its header is included in the editable text. The `To:`, `Cc:`, `Bcc:`, `Subject:`, `From:`, `Reply-To:`, `Sender:`, and `Organization:` fields are accepted within the header, other fields are ignored.

emptystart

If the mailbox is empty S-nail normally prints “No mail for user” and exits immediately. If this option is set S-nail starts even with an empty mailbox.

flipr

This option reverses the meanings of a set of reply commands, turning the lowercase variants, which by default address all recipients included in the header of a message (**reply**[157], **respond**[163], **followup**[110]) into the uppercase variants, which by default address the sender only (**Reply**[156], **Respond**[162], **Followup**[109]) and vice versa. The commands **replysender**[159], **respondsender**[165], **followupsender**[112] as well as **replyall**[158], **respondall**[164], **followupall**[111] are not affected by the current setting of *flipr*[290].

followup-to

Controls whether a `Mail-Followup-To:` header is generated when sending messages to known mailing lists. Also see *followup-to-honour*[370] and the commands **m**list[134], **m**subscribe[135], **reply**[157] and **L**reply[129].

forward-as-attachment

Original messages are normally sent as inline text with the **forward**[114] command, and only the first part of a multipart message is included. With this option messages are sent as unmodified MIME message/rfc822 attachments with all of their parts included.

fullnames

When replying to a message S-nail normally removes the comment parts of email addresses, which by convention contain the full names of the recipients. If this variable is set such stripping is not performed, and comments are retained.

header

Causes the header summary to be written at startup and after commands that affect the number of messages or the order of messages in the current folder; enabled by default. The command line option **-N**[52] can be used to set *noheader*[294].

history-gabby

[Option] Add more entries to the history as is normally done.

history-gabby-persist

[Option] S-nails own NCL will not save the additional (gabby) history entries in persistent storage unless this variable is also set. Also see *NAIL_HISTFILE*[383].

- hold* This option is used to hold messages in the system mailbox by default.
- idna-disable* [Option] Can be used to turn off the automatic conversion of domain names according to the rules of IDNA (internationalized domain names for applications). Since the IDNA code assumes that domain names are specified with the *tycharset*[453] character set, an UTF-8 locale charset is required to represent all possible international domain names (before conversion, that is).
- ignore* Ignore interrupt signals from the terminal while entering messages; instead echo them as '@' characters and discard the current line.
- ignoreeof* Ignore end-of-file conditions (`control-D`), on message input, which instead can be terminated only by entering a *dot*[286] '.' on a line by itself or by using the '~'. [229] **TILDE ESCAPES**[6]. This option also applies to S-nail command mode.
- keep* If set, an empty mailbox file is not removed. This may improve the interoperability with other mail user agents when using a common folder directory, and prevents malicious users from creating fake mailboxes in a world-writable spool directory. Note this only applies to local regular (MBOX) files, other mailbox types will never be removed.
- keepsave* When a message is saved it is usually discarded from the originating folder when S-nail is quit. Setting this option causes all saved message to be retained.
- line-editor-disable* Turn off any enhanced command line editing capabilities (see **Command line editor**[27] for more).
- markanswered* When a message is replied to and this variable is set, it is marked as having been answered. This mark has no technical meaning in the mail system; it just causes messages to be marked in the header summary, and makes them specially addressable.
- mbx-rfc4155* S-nail produces and expects fully RFC 4155 compliant MBOX text mailboxes. Messages which are fetched over the network or from within already existing Maildir (or any non-MBOX) mailboxes may require so-called `From_` quoting (insertion of additional '>' characters to prevent line content misinterpretation) to be applied in order to be storable in MBOX mailboxes, however, dependent on the circumspection of the message producer. (E.g., S-nail itself will, when newly generating messages, choose a *Content-Transfer-encoding*[362] that prevents the necessity for such quoting – a necessary precondition to ensure message checksums won't change.)
By default S-nail will perform this `From_` quoting in a way that results in a MBOX file that is compatible with the loose POSIX MBOX layout, in order not to exceed the capabilities of simple applications, however. Set this option to generate MBOX files for RFC 4155 compliant applications only.
- message-id-disable* By setting this option the generation of `Message-ID:` can be completely suppressed, effectively leaving this task up to the mail-transfer-agent (MTA) or the SMTP server. (According to RFC 5321 your SMTP server is not required to add this field by itself, so you should ensure that it accepts messages without a `Message-ID`.)
- metoo* Usually, when an **alias**[77] expansion contains the sender, the sender is removed from the expansion. Setting this option suppresses these removals. Note that a *setmetoo* [307] also causes a '-m' option to be passed to mail-transfer-agents (MTAs); though most of the modern MTAs don't (no longer) document this flag, no MTA is known which doesn't support it (for historical compatibility).

mime-allow-text-controls

When sending messages, each part of the message is MIME-inspected in order to classify the `Content-Type:` and `Content-Transfer-Encoding:` (see *encoding[362]*) that is required to send this part over mail transport, i.e., a computation rather similar to what the `file(1)` command produces when used with the `--mime` option.

This classification however treats text files which are encoded in UTF-16 (seen for HTML files) and similar character sets as binary octet-streams, forcefully changing any `text/plain` or `text/html` specification to `application/octet-stream`: If that actually happens a yet unset `charset` MIME parameter is set to `binary`, effectively making it impossible for the receiving MUA to automatically interpret the contents of the part.

If this option is set, and the data was unambiguously identified as text data at first glance (by a `.txt` or `.html` file extension), then the original `Content-Type:` will not be overwritten.

netrc-lookup-USER@HOST, netrc-lookup-HOST, netrc-lookup

[v15-compat] [Option] Used to control usage of the users `.netrc` file for lookup of account credentials, as documented in the section **On URL syntax and credential lookup[26]** and for the command **`netrc[140]`**; the section **The `.netrc` file[33]** documents the file format.

outfolder Causes the filename given in the *record[401]* variable and the sender-based filenames for the **`Copy[86]`** and **`Save[167]`** commands to be interpreted relative to the directory given in the *folder[367]* variable rather than to the current directory, unless it is set to an absolute pathname.

page If set, each message the **`pipe[147]`** command prints out is followed by a formfeed character `'\f'`.

piperau Send messages to the **`pipe[147]`** command without performing MIME and character set conversions.

pop3-bulk-load-USER@HOST, pop3-bulk-load-HOST, pop3-bulk-load

[Option] When accessing a POP3 server S-nail loads the headers of the messages, and only requests the message bodies on user request. For the POP3 protocol this means that the message headers will be downloaded twice. If this option is set then S-nail will download only complete messages from the given POP3 server(s) instead.

pop3-no-apop-USER@HOST, pop3-no-apop-HOST[315], pop3-no-apop

[Option] Unless this variable is set the APOP authentication method will be used when connecting to a POP3 server that advertises support. The advantage of APOP is that the password is not sent in clear text over the wire and that only a single packet is sent for the user/password tuple. Note that *pop3-no-apop-HOST* requires [v15-compat].

pop3-use-starttls-USER@HOST, pop3-use-starttls-HOST, pop3-use-starttls

[Option] Causes S-nail to issue a STLS command to make an unencrypted POP3 session SSL/TLS encrypted. This functionality is not supported by all servers, and is not used if the session is already encrypted by the POP3S method. Note that *pop3-use-starttls-HOST* requires [v15-compat].

print-all-chars

This option causes all characters to be considered printable. It is only effective if given in a startup file. With this option set some character sequences in messages may put the user's terminal in an undefined state when printed; it should only be used as a last resort if no working system locale can be found.

print-alternatives

When a MIME message part of type `multipart/alternative` is displayed and it contains a subpart of type `text/plain`, other parts are normally discarded. Setting this variable causes

all subparts to be displayed, just as if the surrounding part was of type `multipart/mixed`.

quiet Suppresses the printing of the version when first invoked.

quote-as-attachment

If this is set, then the original message is added in its entirety as a `message/rfc822` MIME attachment when replying to a message. Note this works regardless of the setting of `quote[399]`.

recipients-in-cc

On group replies, specify only the sender of the original mail in `To:` and mention the other recipients in the secondary `Cc:`. By default all recipients of the original mail will be addressed via `To:`.

record-resent

If both this variable and the `record[401]` variable are set, the `resend[161]` and `Resend[160]` commands save messages to the `record[401]` folder as it is normally only done for newly composed messages.

reply-in-same-charset

If this variable is set S-nail first tries to use the same character set of the original message for replies. If this fails, the mechanism described in **Character sets[23]** is evaluated as usual.

rfc822-body-from_

This variable can be used to force displaying a so-called `From_` line for messages that are embedded into an envelope mail via the `message/rfc822` MIME mechanism, for more visual convenience.

save When the user aborts a message with two “RUBOUT” (interrupt,control-C) characters, S-nail will copy the partial letter to the file `DEAD[460]`.

searchheaders

Expand message-list specifiers in the form `/x:y` to all messages containing the substring “y” in the header field ‘x’. The string search is case insensitive.

sendcharsets-else-ttycharset

[Option] If this variable is set, but `sendcharsets[406]` is not, then S-nail acts as if `sendcharsets[406]` had been set to the value of the variable `ttycharset[453]`. In effect this combination passes through the message data in the character set of the current locale (given that `ttycharset[453]` hasn’t been set manually), i.e., without converting it to the `charset-8bit[349]` fallback character set. Thus, mail message text will be in ISO-8859-1 encoding when send from within a ISO-8859-1 locale, and in UTF-8 encoding when send from within an UTF-8 locale. If no character set conversion capabilities are available in S-nail then the only supported character set is `ttycharset[453]`.

sendwait When sending a message wait until the MTA (including the builtin SMTP one) exits before accepting further commands. **Only** with this variable set errors reported by the MTA will be recognizable! If the MTA returns a non-zero exit status, the exit status of s-nail will also be non-zero.

showlast Setting this option causes S-nail to start at the last message instead of the first one when opening a mail folder.

showname

Causes S-nail to use the sender’s real name instead of the plain address in the header field summary and in message specifications.

- showto* Causes the recipient of the message to be shown in the header summary if the message was sent by the user.
- skipemptybody*
If an outgoing message does not contain any text in its first or only message part, do not send it but discard it silently (see also the command line option **-E** [44]).
- smime-force-encryption*
[Option] Causes S-nail to refuse sending unencrypted messages.
- smime-sign*
[Option] S/MIME sign outgoing messages with the user's private key and include the user's certificate as a MIME attachment. Signing a message enables a recipient to verify that the sender used a valid certificate, that the email addresses in the certificate match those in the message header and that the message content has not been altered. It does not change the message text, and people will be able to read the message as usual. Also see *smime-sign-cert*[420] and *smime-sign-include-certs*[421].
- smime-no-default-ca*
[Option] Don't load default CA locations when verifying S/MIME signed messages.
- smtp-use-starttls-USER@HOST, smtp-use-starttls-HOST, smtp-use-starttls*
[Option] Causes S-nail to issue a `STARTTLS` command to make an SMTP session SSL/TLS encrypted, i.e., to enable transport layer security.
- ssl-no-default-ca*
[Option] Don't load default CA locations to verify SSL/TLS server certificates.
- term-ca-mode*
[Option] If terminal capability queries are supported and this option is set then S-nail will try to create a terminal session when in interactive mode, which will cause a screen clearance once S-nail is terminated again. **Note:** even when supported for this to produce appealing results the used `PAGER`[474] and possibly configured *pipe-TYPE/SUBTYPE*[390] applications that take control over the terminal need to have corresponding support too, e.g., the `less(1)` pager should be driven with the `'-X'` command line flag.
- keep-content-length*
When (editing messages and) writing `MBOX`[470] mailbox files S-nail can be told to keep the `Content-Length:` and `Lines:` header fields that some MUAs generate by setting this variable. Since S-nail does neither use nor update these non-standardized header fields (which in itself shows one of their conceptual problems), stripping them should increase interoperability in between MUAs that work with with same mailbox files. Note that, if this is not set but *writebackedited*[342], as below, is, a possibly performed automatic stripping of these header fields already marks the message as being modified.
- v15-compat*
Setting this option enables upward compatibility with S-nail version 15.0 in respect to which configuration options are available and how they are handled. This manual uses `[v15-compat]` and `[no v15-compat]` to refer to the new and the old way of doing things, respectively.
- verbose* Setting this option, also controllable via the command line option **-v** [62], causes S-nail to be more verbose, so that, e.g., certificate chains will be displayed on the users terminal. Setting this binary option twice increases the level of verbosity, in which case even details of the actual message delivery and protocol conversations are shown. A single *noverbose*[341] is sufficient to disable verbosity as such.

writebackedited

If this variable is set messages modified using the `edit`[98] or `visual`[222] commands are written back to the current folder when it is quit; it is only honoured for writable folders in MBOX format, though. Note that the editor will be pointed to the raw message content in that case, i.e., neither MIME decoding nor decryption will have been performed, and proper RFC 4155 `From_` quoting of newly added or edited content is also left as an exercise to the user.

Value options[31]

Options with values that are generally treated as strings. To embed whitespace (space and tabulator) in a value it either needs to be escaped with a backslash character, or the entire value must be enclosed in (double or single) quotation marks; To use quotation marks identical to those used to enclose the value, escape them with a backslash character. The backslash character has no special meaning except in these cases.

```
set 1=val\ one 2="val two" 3='val "three"' 4='val \'four\''
var 1 2 3 4
unset 1 2 3 4
```

Booleans are special string values that must either be set to decimal integers (in which case ‘0’ is false and ‘1’ and any other value is true) or to any of `off`, ‘no’ and `false` for a false boolean and ‘on’, `yes` and `true` for a true boolean; matching is performed case-insensitively. And there exists a special kind of boolean, the “quadoption”: this is expected to either name a boolean or one of the strings `ask-yes` and `ask-no` (in fact: `ask-` followed by a valid boolean, case-insensitively); if one of the latter is set then in interactive mode the user will be prompted with the default value (also used for empty user input) set to the given boolean, whereas in non-interactive the given default will be used right away.

agent-shell-lookup-USER@HOST, agent-shell-lookup-HOST, agent-shell-lookup

[v15-compat] [Option] Account passwords can be fetched via an external agent program in order to permit encrypted password storage – see **On URL syntax and credential lookup**[26] for more on credential lookup. If this is set then the content is interpreted as a shell command the output of which (with newline characters removed) is treated as the account password shall the command succeed (and have produced non-empty non-newline output); e.g., via `gpg`(1):

```
$ echo PASSWORD > .pass
$ gpg -e .pass
$ eval `gpg-agent --daemon \
  --pinentry-program=/usr/bin/pinentry-curses \
  --max-cache-ttl 99999 --default-cache-ttl 99999`
$ echo 'set agent-shell-lookup="gpg -d .pass.gpg"' \
>> ~/.mailrc
```

A couple of environment variables will be set for the agent:

<code>NAIL_TMPDIR</code> [391]	The temporary directory that S-nail uses. Usually identical to <code>TMPDIR</code> [479], but guaranteed to be set and usable by child processes; to ensure the latter condition for <code>TMPDIR</code> [479] also, it’ll be set.
<code>NAIL_USER</code>	The user (USER) for which the password is looked up.
<code>NAIL_USER_ENC</code>	The URL percent-encoded variant of <code>NAIL_USER</code> .
<code>NAIL_HOST</code>	The plain machine hostname of the user account.
<code>NAIL_HOST_PORT</code>	The <code>HOST</code> (hostname possibly including port) of the user account.

attrlist

A sequence of characters to print in the `attribute` column of the `headline`[373] as shown in the header display; each for one type of messages (see **Message states**[24]), with the default being `NUROSPMFAT+-$~` or `NU *HMFAT+-$~` if `bsdflags`[277] or the `SYSV3`[477] environment variable are set, in the following order:

‘N’ new.
 ‘U’ unread but old.
 ‘R’ new but read.
 ‘O’ read and old.
 ‘S’ saved.
 ‘P’ preserved.
 ‘M’ mboxed.
 ‘F’ flagged.
 ‘A’ answered.
 ‘T’ draft.
 ‘+’ start of a collapsed thread.
 ‘-’ collapsed.
 ‘\$’ classified as spam.
 ‘~’ classified as possible spam.

autobcc Specifies a list of recipients to which a blind carbon copy of each outgoing message will be sent automatically.

autocc Specifies a list of recipients to which a carbon copy of each outgoing message will be sent automatically.

autosort Causes sorted mode (see the **sort**[179] command) to be entered automatically with the value of this option as sorting method when a folder is opened.

charset-7bit

The value that should appear in the `charset=` parameter of `Content-Type:` MIME header fields when no character set conversion of the message data was performed. This defaults to US-ASCII, and the chosen character set should be US-ASCII compatible.

charset-8bit

[Option] The default 8-bit character set that is used as an implicit last member of the variable *sendcharsets*[406]. This defaults to UTF-8. If no character set conversion capabilities are available in S-nail then the only supported character set is *ttycharset*[453]. Refer to the section **Character sets**[23] for the complete picture of character set conversion in S-nail.

charset-unknown-8bit

[Option] RFC 1428 specifies conditions when internet mail gateways shall “upgrade” the content of a mail message by using a character set with the name `unknown-8bit`. Because of the unclassified nature of this character set S-nail will not be capable to convert this character set to any other character set. If this variable is set any message part which uses the character set `unknown-8bit` is assumed to really be in the character set given in the value, otherwise the (final) value of *charset-8bit*[349] is used for this purpose.

cmd The default value for the **pipe**[147] command.

colour-from_

[Option] The colour specification for so-called `From_` lines. See the section **Coloured message display**[28] for the format of the value.

colour-header

[Option] The colour specification for header lines.

colour-msginfo

[Option] The colour specification for the introductory message info line.

colour-partinfo

[Option] The colour specification for MIME part info lines.

colour-terms

[Option] A comma-separated list of `TERM`[478]inals for which coloured message display can be used. Entries only need to be added if the string “color” isn’t part of the terminal name itself; the default value is

```
cons25,linux,rxvt,rxvt-unicode,screen,sun,vt100,vt220,
wsvt25,xterm
```

colour-uheader

[Option] The colour specification for those header lines that have been placed in the *colour-user-headers*[358] list. See the section **Coloured message display**[28].

colour-user-headers

A comma separated list of (case-insensitive) header names which should be colourized with the alternative *colour-uheader*[357] colours. The default value is `from,subject`.

crt

The valued option `crt` is used as a threshold to determine how long a message must be before `PAGER`[474] is used to read it, where the value ‘0’ means to always use it. If *crt*[359] is set without a value then the height of the terminal screen stored in the system is used to compute the threshold (see `LINES`[468] and `stty(1)`).

datefield

The date in a header summary is normally the date of the mailbox `From_` line of the message. If this variable is set, then the date as given in the `Date:` field is used, converted to local time. It is possible to control the display of the date by assigning a value, in which case the `strftime(3)` function will be used to format the date accordingly. Please read your system manual for the available formats. Note that the ‘%n’ format should not be used, because S-nail doesn’t take embedded newlines into account when calculating how many lines fit onto the screen.

datefield-markout-older

This option, when set in addition to *datefield*[360], is used to display “older” messages (concept is rather comparable to the `-l` option of the POSIX utility `ls(1)`). The content interpretation is identical to *datefield*[360].

encoding

Suggestion for the MIME encoding to use in outgoing text messages and message parts. Valid values are the default `quoted-printable`, `8bit` and `base64`. `8bit` may cause problems when transferring mail messages over channels that are not ESMTP (RFC 1869) compliant. If there is no need to encode a message, `7bit` transfer mode is always used regardless of this variable. Binary data is always encoded as `base64`.

escape

If defined, the first character of this option gives the character to use in place of ‘~’ to denote **TILDE ESCAPES**[6].

expandaddr

If set then an extended way of interpreting recipient address specifications is used – see the section **Sending mail**[18] for more on this. To disallow any extended interpretation only in non-interactive mode, unless tilde commands were enabled explicitly by using one of the command line options `-~` [64] or `-#` [65], set this to the (case-insensitive) value “restrict”. If set to the value “fail” then the existence of extended recipient addresses is treated as a hard send error (instead of only filtering them out).

In fact the value is interpreted as a comma-separated list of strings. To forbid that anything that is not a file or pipe addressee (and thus labelled as “extended recipient address” around here) will be treated as a network address, but to filter out aliases, too, include the value “noalias” (which will be a hard error only in conjunction with “fail”); note this applies to system aliases only,

S-nail internal **alias**[77]es are not affected by this setting (only their expansion).

expandargv

Unless this variable is set additional mail-transfer-agent (MTA) arguments from the command line, as can be given after a ‘--’ separator, are ignored due to safety reasons. However, if set to the special value “fail”, then the presence of additional MTA arguments is treated as a hard error that causes S-nail to exit with failure status. A lesser strict variant is the otherwise identical “restrict”, which does accept such arguments in interactive mode, or if tilde commands were enabled explicitly by using one of the command line options `-~` [64] or `-#` [65].

features

Information on the features compiled into S-nail – the content of this variable is identical to the output of the command **features**[103].

folder

The name of the directory to use for storing folders of messages. All folder names that begin with ‘+’ refer to files below it. The same special conventions as documented for the **file**[105] command may be used when specifying a new value for *folder*, but be aware that the expansion is fully performed immediately.

folder-hook

When a folder is opened and this variable is set, the macro corresponding to the value of this variable is executed. The macro is also invoked when new mail arrives, but message lists for commands executed from the macro only include newly arrived messages then. If **localopts**[128] are activated in a folder hook, then the covered settings will be reverted once the folder is left again.

folder-hook-FOLDER

Overrides *folder-hook*[368] for a folder named FOLDER. Unlike other folder specifications, the fully expanded name of a folder, without metacharacters, is used to avoid ambiguities. However, if the mailbox resides under *folder*[367] then the usual ‘+’ specification is tried in addition, e.g., if *folder* is “mail” (and thus relative to the user’s home directory) then /home/usr1/mail/sent will be tried as *folder-hook*-/home/usr1/mail/sent first, but then followed by *folder-hook*-+sent.

followup-to-honour

Controls whether a Mail-Followup-To: header is honoured when group-replying to a message via **reply**[157] or **Lreply**[129]. This is a quadoption; if set without a value it defaults to “yes”. Also see *followup-to*[291] and the commands **mlist**[134] and **mlsubscribe**[135].

from

The address (or a list of addresses) to put into the From: field of the message header, quoting RFC 5322: the author(s) of the message, that is, the mailbox(es) of the person(s) or system(s) responsible for the writing of the message. If replying to messages these addresses are handled as if they were in the **alternates**[78] list. If the machine’s hostname is not valid at the Internet (for example at a dialup machine) then either this variable or *hostname*[375] ([v15-compat] and with *smtp*[422] *smtp-hostname*[424] adds even more fine-tuning capabilities), have to be set. If *from* contains more than one address, setting the *sender*[407] variable is required (according to the standard RFC 5322).

fwdheading

The string to print before the text of a message with the **forward**[114] command (unless the *forward-as-attachment*[292] variable is set). Defaults to “----- Original Message -----” if unset; No heading is printed if it is set to the empty string.

headline

A format string to use for the header summary, similar to `printf(3)` formats. A percent character ‘%’ introduces a format specifier that may be followed by a number indicating the field width; If the (possibly implicitly implied) field width is negative, the field is to be left-aligned. Valid

format specifiers are:

- '%%' A plain percent character.
- '%>' A space character but for the current message, for which it expands to '>'.
- '%<' A space character but for the current message, for which it expands to '<'.
- '%\$' [Option] The spam score of the message, as has been classified via the command **spamrate**[184]. Prints only a replacement character if there is no spam support.
- '%a' Message attribute character (status flag); the actual content can be adjusted by setting *attrlist*[344].
- '%d' The date when the message was received.
- '%e' The indenting level in threaded mode.
- '%f' The address of the message sender.
- '%i' The message thread structure. (Note that this format doesn't support a field width.)
- '%l' The number of lines of the message.
- '%m' Message number.
- '%o' The number of octets (bytes) in the message.
- '%s' Message subject (if any).
- '%S' Message subject (if any) in double quotes.
- '%T' Message recipient flags: is the addressee of the message a known or subscribed mailing list – see **mlist**[134] and **mlsubscribe**[135].
- '%t' The position in threaded/sorted order.

The default is `%>%a%m %-18f %16d %4l/%-5o %i%-s,` or `%>%a%m %20-f %16d %3l/%-5o %i%-S` if *bsdcompat*[276] is set. Also see *attrlist*[344] and *headline-bidi*[374].

headline-bidi

Bidirectional text requires special treatment when displaying headers, because numbers (in dates or for file sizes etc.) will not affect the current text direction, in effect resulting in ugly line layouts when arabic or other right-to-left text is to be displayed. On the other hand only a minority of terminals is capable to correctly handle direction changes, so that user interaction is necessary for acceptable results. Note that extended host system support is required nonetheless, e.g., detection of the terminal character set is one precondition; and this feature only works in an Unicode (i.e., UTF-8) locale.

In general setting this variable will cause S-nail to encapsulate text fields that may occur when printing *headline*[373] (and some other fields, like dynamic expansions in *prompt*[398]) with special Unicode control sequences; it is possible to fine-tune the terminal support level by assigning a value: no value (or any value other than '1', '2' and '3') will make S-nail assume that the terminal is capable to properly deal with Unicode version 6.3, in which case text is embedded in a pair of U+2068 (FIRST STRONG ISOLATE) and U+2069 (POP DIRECTIONAL ISOLATE) characters. In addition no space on the line is reserved for these characters.

Weaker support is chosen by using the value '1' (Unicode 6.3, but reserve the room of two spaces for writing the control sequences onto the line). The values '2' and '3' select Unicode 1.1 support (U+200E, LEFT-TO-RIGHT MARK); the latter again reserves room for two spaces in addition.

hostname Use this string as hostname when expanding local addresses instead of the value obtained from `uname(3)` and `getaddrinfo(3)`, i.e., in `Message-ID:` and `From:` fields. Note that when *smtp*[422] transport is not used then it is normally the responsibility of the MTA to create these fields, [v15-compat] in conjunction with *smtp*[422] however *smtp-hostname*[424] also influences the results; you should produce some test messages with the desired combination of *hostname*,

and/or *from*[371], *sender*[407] etc. first.

indentprefix

String used by the `~m`[247], `~M`[246] and `~R`[250] **TILDE ESCAPES**[6] and by the *quote*[399] option for indenting messages, in place of the normal tabulator character ‘`^I`’, which is the default. Be sure to quote the value if it contains spaces or tabs.

line-editor-cursor-right

[Option] If the builtin command line editor is used, actions which are based on rightwise movement may not work on some terminals. If you encounter such problems, set this variable to the terminal control sequence that is necessary to move the cursor one column to the right. The default is `[C`, which should work for most terminals. Less often occur `OC` and `.`. Note that “Escape” and other control character have to be written as shell-style escape sequences, e.g., `^C` for (US-ASCII) “Escape”.

MAIL

Is used as the user’s mailbox, if set. Otherwise, a system-dependent default is used. Supports a logical subset of the special conventions that are documented for the *file*[105] command and the *folder*[367] option.

mime-counter-evidence

Normally the `Content-Type:` field is used to decide how to handle MIME parts. Some MUAs however don’t use `mime.types(5)` or a similar mechanism to correctly classify content, but simply specify `application/octet-stream`, even for plain text attachments like `text/diff`. If this variable is set then S-nail will try to classify such MIME message parts on its own, if possible, and through their file name. This variable can also be given a non-empty value, in which case the value is expected to be a number, actually a carrier of bits. Creating the bit-carrying number is a simple addition:

```
? !echo Value should be set to  $$(2 + 4)$ 
Value should be set to 6
```

- If bit two is set (2) then the detected content-type will be carried along with the message and be used for deciding which *pipe-TYPE/SUBTYPE*[390] is responsible for the MIME part, shall that question arise; when displaying such a MIME part the part-info will indicate the overridden content-type by showing a plus-sign ‘+’.
- If bit three is set (4) then the counter-evidence is always produced and a positive result will be used as the MIME type, even forcefully overriding the parts given MIME type.

mimetypes-load-control

This option can be used to control which of the `mime.types(5)` databases are loaded by S-nail, as furtherly described in the section **The mime.types files**[32]. If the letter ‘u’ is part of the option value, then the user’s personal `~/mime.types`[482] file will be loaded (if it exists); likewise the letter ‘s’ controls loading of the system wide `/etc/mime.types` [483]; the user file is loaded first, letter matching is case-insensitive. If this option is not set S-nail will try to load both files instead. Incorporation of the S-nail-builtin MIME types cannot be suppressed, but they will be matched last.

More sources can be specified by using a different syntax: if the value string contains an equals sign ‘=’ then it is instead parsed as a comma-separated list of the described letters plus `f=FILENAME` pairs; the given filenames will be expanded and loaded, and their content may use the extended syntax that is described in the section **The mime.types files**[32].

NAIL_EXTRA_RC

The name of an optional startup file to be read after `~/mailrc`[68]. This variable has an effect only if it is set in `/usr/local/etc/s-nail.rc`[67] or `~/mailrc`, it is not imported from the environment in order to honour `MAILRC=/dev/null/ -n`[53] invocations.

Use this file for commands that are not understood by other POSIX mailx(1) implementations.

NAIL_HEAD

A string to put at the beginning of each new message. The escape sequences tabulator '\t' and newline '\n' are understood.

NAIL_HISTFILE

[Option] If a command line editor is available then this can be set to name the (expandable) path of the location of a permanent history file.

NAIL_HISTSIZE

[Option] If a command line editor is available this value restricts the amount of history entries that are saved into a set and valid *NAIL_HISTFILE*[383]. A value of less than 0 disables this feature; note that loading and incorporation of *NAIL_HISTFILE*[383] upon program startup can also be suppressed by doing this. An unset or invalid value, or 0, causes a default value to be used. Dependent on the available command line editor this will also define the number of history entries in memory; it is also editor-specific whether runtime updates of this value will be honoured.

NAIL_TAIL

A string to put at the end of each new message. The escape sequences tabulator '\t' and newline '\n' are understood.

newfolders

If this variable has the value *maildir*, newly created local folders will be in Maildir format.

newmail

Checks for new mail in the current folder each time the prompt is printed. A Maildir folder must be re-scanned to determine if new mail has arrived. If this variable is set to the special value *nopoll* then a Maildir folder will not be rescanned completely, but only timestamp changes are detected.

ORGANIZATION

The value to put into the *Organization:* field of the message header.

password-USER@HOST, password-HOST, password

[v15-compat] Variable chain that sets a password, which is used in case none has been given in the protocol and account-specific URL; as a last resort S-nail will ask for a password on the user's terminal if the authentication method requires a password. Specifying passwords in a startup file is generally a security risk; the file should be readable by the invoking user only.

password-USER@HOST

[no v15-compat] (see the chain above for [v15-compat]) Set the password for *USER* when connecting to *HOST*. If no such variable is defined for a host, the user will be asked for a password on standard input. Specifying passwords in a startup file is generally a security risk; the file should be readable by the invoking user only.

pipe-TYPE/SUBTYPE

When a MIME message part of type *TYPE/SUBTYPE* (normalized to lowercase) is displayed or quoted, its text is filtered through the value of this variable interpreted as a shell command. The special value '@' can be used to force interpretation of the message part as plain text, e.g., *set pipe-application/pgp-signature=@* will henceforth treat signatures as plain text and display them "as is".

Also, if a shell command is prefixed with '@', then the command will only be used to prepare the MIME message part if the message is displayed by itself, but not when multiple messages are displayed at once.

Finally, if a shell command is prefixed with '@&', then, in addition to what has been described for the plain '@' shell command prefix, the command will be run asynchronously, i.e., without blocking S-nail, which may be a handy way to display a, e.g., PDF file while also continuing to read the mail message. Some information about the MIME part to be displayed is embedded into the environment of the shell command:

NAIL_TMPDIR	The temporary directory that S-nail uses. Usually identical to TMPDIR[479], but guaranteed to be set and usable by child processes; to ensure the latter condition for TMPDIR also, it'll be set.
NAIL_FILENAME	The filename, if any is set, the empty string otherwise.
NAIL_FILENAME_GENERATED	Set to NAIL_FILENAME if that isn't empty, but otherwise a combination of a random string (always) and the SUBTYPE of the MIME part, if the latter is known.
NAIL_CONTENT	The MIME content-type of the part, if known, the empty string otherwise.
NAIL_CONTENT_EVIDENCE	If <i>mime-counter-evidence</i> [379] includes the carry-around-bit (2), then this will be set to the detected MIME content-type; not only then identical to NAIL_CONTENT otherwise.

pipe-EXTENSION

This is identical to *pipe-TYPE/SUBTYPE*[390] except that EXTENSION (normalized to lower-case using character mappings of the ASCII charset) names a file extension, e.g., xhtml. Handlers registered using this method take precedence.

pop3-keepalive

[Option] POP3 servers close the connection after a period of inactivity; the standard requires this to be at least 10 minutes, but practical experience may vary. Setting this variable to a numeric value greater than '0' causes aNOOP command to be sent each value seconds if no other operation is performed.

prompt

The string printed when a command is accepted. Prompting may be prevented by either setting this to the null string or by setting *noprompt*[398]. The same XSI escape sequences that are understood by the **echo**[97] command may be used within *prompt*.

In addition, the following S-nail specific additional sequences are understood: '\&', which expands to "?" unless *bsdcompat*[276] is set, in which case it expands to "&"; note that \& is the default value of *prompt*. '\?', which will expand to "1" if the last command failed and to "0" otherwise, '\\$', which will expand to the name of the currently active **account**[76], if any, and to the empty string otherwise, and '\@', which will expand to the name of the currently active mailbox. (Note that the prompt buffer is size-limited, excess is cut off.)

Even though *prompt* checks for *headline-bidi*[374] to encapsulate the expansions of the '\\$' and '\@' escape sequences as necessary to correctly display bidirectional text, this is not true for the final string that makes up *prompt* as such, i.e., real BIDI handling is not supported.

When a newer version of the *editline*(3) **Command line editor**[27] is used, any escape sequence must itself be encapsulated with another escape character for usage with the *EL_PROMPT_ESC* mechanism: S-nail configures the control character \01 for this.

quote

If set, S-nail starts a replying message with the original message prefixed by the value of the variable *indentprefix*[376]. Normally, a heading consisting of "Fromheaderfield wrote:" is printed before the quotation. If the string *noheading* is assigned to the *quote* variable, this heading is omitted. If the *stringheaders* is assigned, the headers selected by the **ignore**[126]/**retain**[166] commands are printed above the message body, thus *quote* acts like an automatic '~m[247] **TILDE ESCAPES**[6] command, then. If the string

`allheaders` is assigned, all headers are printed above the message body and all MIME parts are included, making `quote` act like an automatic ‘~**M**[246]’ command; also see `quote-as-attachment`[320].

`quote-fold` [Option] Can be set in addition to `indentprefix`[376]. Setting this turns on a more fancy quotation algorithm in that leading quotation characters are compressed and overlong lines are folded. `quote-fold` can be set to either one or two (space separated) numeric values, which are interpreted as the maximum (goal) and the minimum line length, respectively, in a spirit rather equal to the `fmt(1)` program, but line-, not paragraph-based. If not set explicitly the minimum will reflect the goal algorithmically. The goal can’t be smaller than the length of `indentprefix`[376] plus some additional pad. Necessary adjustments take place silently.

`record` If defined, gives the pathname of the folder used to record all outgoing mail. If not defined, then outgoing mail is not saved. When saving to this folder fails the message is not sent, but instead saved to `DEAD`[460].

`reply_strings`

Can be set to a comma-separated list of (case-insensitive according to ASCII rules) strings which shall be recognized in addition to the builtin strings as `Subject: reply` message indicators – builtin are `Re:`, which is mandated by RFC 5322, as well as the german `Aw:`.

`replyto` A list of addresses to put into the `Reply-To:` field of the message header. Members of this list are handled as if they were in the `alternates`[78] list.

`reply-to-honour`

Controls whether a `Reply-To:` header is honoured when replying to a message via `reply`[157] or `Lreply`[129]. This is a quadoption; if set without a value it defaults to “yes”.

`screen` When S-nail initially prints the message headers it determines the number to print by looking at the speed of the terminal. The faster the terminal, the more it prints. This option overrides this calculation and specifies how many message headers are printed. This number is also used for scrolling with the `z`[225] command.

`sendcharsets`

[Option] A comma-separated list of character set names that can be used in outgoing internet mail. The value of the variable `charset-8bit`[349] is automatically appended to this list of character-sets. If no character set conversion capabilities are compiled into S-nail then the only supported charset is `tycharset`[453]. Also see `sendc harsets-else-tycharset`[327] and refer to the section **Character sets**[23] for the complete picture of character set conversion in S-nail.

`sender` An address that is put into the `Sender:` field of outgoing messages, quoting RFC 5322: the mailbox of the agent responsible for the actual transmission of the message. This field should normally not be used unless the `From:` field contains more than one address, on which case it is required. The `sender` address is handled as if it were in the `alternates` [78] list.

`sendmail` To use an alternate mail transport agent (MTA), set this option to the full pathname of the program to use. It may be necessary to set `sendmail-progname`[410] in addition.

The MTA will be passed command line arguments from several possible sources: from the variable `sendmail-arguments`[409] if set, from the command line if given and the variable `expandargv`[365] allows their use. It’ll always be passed the ‘-i’ option, the option ‘-m’ only if the `metoo`[307] variable, and ‘-v’ only if the `verbose` [341] variable are set. In conjunction with the `-r`[56] command line option S-nail may pass `-f` as well as `-F`. Option processing of the MTA will be terminated via a ‘--’ separator.

sendmail-arguments

Arguments to pass through to the Mail-Transfer-Agent can be given via this option. The content of this variable will be split up in a vector of arguments which will be joined onto other possible MTA options:

```
set sendmail-arguments='-t -X "/tmp/my log"'
```

sendmail-progname

Many systems use a so-called mailwrapper(8) environment to ensure compatibility with sendmail(1). This works by inspecting the name that was used to invoke the mail delivery system. If this variable is set then the mailwrapper (the program that is actually executed when calling “sendmail”) will treat its contents as that name. The default is `sendmail`.

Sign A string for use with the `~A[235]` tilde escape.

sign A string for use with the `~a[236]` tilde escape.

signature Must correspond to the name of a readable file if set. The file’s content is then appended to each singlepart message and to the first part of each multipart message. Be warned that there is no possibility to edit the signature for an individual message.

smime-ca-dir

[Option] Specifies a directory with CA certificates in PEM (Privacy Enhanced Mail) format for verification of S/MIME signed messages.

smime-ca-file

[Option] Specifies a file with CA certificates in PEM format for verification of S/MIME signed messages.

smime-cipher-USER@HOST

[Option] Specifies a cipher to use when generating S/MIME encrypted messages for the specified account. RFC 5751 mandates a default of “aes-128” (AES-128 CBC). Possible values are (case-insensitive and) in decreasing cipher strength: “aes-256” (AES-256 CBC), “aes-192” (AES-192 CBC), “aes-128” (AES-128 CBC), “des3” (DES EDE3 CBC, 168 bits; default if “aes-128” isn’t available) and “des” (DES CBC, 56 bits).

The actually available cipher algorithms depend on the cryptographic library that S-nail uses. [Option] Support for more cipher algorithms may be available through dynamic loading via, e.g., `EVP_get_cipherbyname(3)` (OpenSSL) if S-nail has been compiled to support this.

smime-crl-file

[Option] Specifies a file that contains a CRL in PEM format to use when verifying S/MIME messages.

smime-crl-dir

[Option] Specifies a directory that contains files with CRLs in PEM format to use when verifying S/MIME messages.

smime-encrypt-USER@HOST

[Option] If this variable is set, messages send to the given receiver are encrypted before sending. The value of the variable must be set to the name of a file that contains a certificate in PEM format.

If a message is sent to multiple recipients, each of them for whom a corresponding variable is set will receive an individually encrypted message; other recipients will continue to receive the message in plain text unless the *smime-force-encryption*[333] variable is set. It is recommended to sign encrypted messages, i.e., to also set the *smime-sign*[334] variable.

smime-sign-cert-USER@HOST, smime-sign-cert

[Option] Points to a file in PEM format. For the purpose of signing and decryption this file needs to contain the user's private key as well as his certificate.

For message signing `USER@HOST` is always derived from the value of `from`[371] (or, if that contains multiple addresses, `sender`[407]). For the purpose of encryption the recipient's public encryption key (certificate) is expected; the command `certsave`[82] can be used to save certificates of signed messages (the section **Signed and encrypted messages with S/MIME**[35] gives some details). This mode of operation is usually driven by the specialized form.

When decrypting messages the account is derived from the recipient fields (`To:` and `Cc:`) of the message, which are searched for addresses for which such a variable is set. S-nail always uses the first address that matches, so if the same message is sent to more than one of the user's addresses using different BSD encryption keys, decryption might fail.

smime-sign-include-certs-USER@HOST, smime-sign-include-certs

[Option] If used, this is supposed to consist of a comma-separated list of files, each of which containing a single certificate in PEM format to be included in the S/MIME message in addition to the *smime-sign-cert*[420] certificate. This is most useful for long certificate chains if it is desired to aid the receiving party's verification process. Note that top level certificates may also be included in the chain but don't play a role for verification. Also see *smime-sign-cert* [420].

smtp

[Option] Normally S-nail invokes the program defined via *sendmail*[408] to transfer messages, as described in **Sending mail**[18]. Setting the *smtp* variable will instead cause SMTP network connections be made to the server specified therein in order to directly submit the message. S-nail knows about three different "SMTP protocols":

- The plain SMTP protocol (RFC 5321) that normally lives on the server port 25 and requires setting the *smtp-use-starttls*[336] variable to enter a SSL/TLS encrypted session state. Assign a value like `[v15-compat] [smtp://][user[:password]@]server[:port]` (`[no v15-compat] [smtp://]server[:port]`) to choose this protocol.
- Then the so-called SMTPS which is supposed to live on server port 465 and is automatically SSL/TLS secured. Unfortunately it never became a standardized protocol and may thus not be supported by your hosts network service database – in fact the port number has already been reassigned to other protocols!

SMTPS is nonetheless a commonly offered protocol and thus can be chosen by assigning a value like `[v15-compat] smtps://[user[:password]@]server[:port]` (`[no v15-compat] smtps://server[:port]`); due to the mentioned problems it is usually necessary to explicitly specify the port as `:465`, however.

- Finally there is the SUBMISSION protocol (RFC 6409), which usually lives on server port 587 and is practically identically to the SMTP protocol from S-nails point of view beside that; it requires setting the *smtp-use-starttls*[336] variable to enter a SSL/TLS secured session state. Assign a value like `[v15-compat] submission://[user[:password]@]server[:port]` (`[no v15-compat] submission://server[:port]`).

For more on credentials etc. please see **On URL syntax and credential lookup**[26]. The SMTP transfer is executed in a child process, which runs asynchronously unless either the *sendwait*[328] or the *verbose*[341] variable is set. If it receives a TERM signal, it will abort and save the message to `DEAD`[460].

smtp-auth-USER@HOST, smtp-auth-HOST, smtp-auth

[Option] Variable chain that sets the SMTP authentication method. Possible values are `none` (`[no v15-compat] default`), `plain` (`[v15-compat] default`), `login` as well as the [Option]al methods `cram-md5` and `gssapi`. Thenone method doesn't need any user credentials,

`gssapi` requires a user name and all other methods require a user name and a password. See `[v15-compat] smtp[422]`, `user[454]` and `password[389]` ([no v15-compat] `smtp-auth-password` and `smtp-auth-user`). Note that `smtp-auth-HOST` is [v15-compat]. [no v15-compat]; Note for `smtp-auth-USER@HOST`: may override dependend on sender address in the variable `from[371]`.

smtp-auth-password

[Option] [no v15-compat] Sets the global fallback password for SMTP authentication. If the authentication method requires a password, but neither `smtp-auth-password` nor a matching `smtp-auth-password-USER@HOST` can be found, S-nail will ask for a password on the user's terminal.

smtp-auth-password-USER@HOST

[no v15-compat] Overrides `smtp-auth-password` for specific values of sender addresses, dependent upon the variable `from[371]`.

smtp-auth-user

[Option] [no v15-compat] Sets the global fallback user name for SMTP authentication. If the authentication method requires a user name, but neither `smtp-auth-user` nor a matching `smtp-auth-user-USER@HOST` can be found, S-nail will ask for a user name on the user's terminal.

smtp-auth-user-USER@HOST

[no v15-compat] Overrides `smtp-auth-user` for specific values of sender addresses, dependent upon the variable `from[371]`.

smtp-hostname

[v15-compat] Normally S-nail uses the variable `from[371]` to derive the necessary `USER@HOST` information to issue a `MAIL FROM:<> SMTP` command. Setting `smtp-hostname[424]` can be used to use the `USER` from the SMTP account (`smtp[422]` or the `user[454]` variable chain) and the `HOST` from the content of this variable (or, if that is the empty string, `hostname[375]` or the local hostname as a last resort). This often allows using an address that is itself valid but hosted by a provider other than which (in `from[371]`) is about to send the message. Setting this variable also influences the generated `Message-ID:`.

spam-interface

[Option] In order to use any of the spam-related commands (like, e.g., `spamrate[184]`) the desired spam interface must be defined by setting this variable. Please refer to the manual section **Handling spam[37]** for the complete picture of spam handling in S-nail. All or none of the following interfaces may be available:

`spamc` Interaction with `spamc(1)` from the `spamassassin(1)` (<http://spamassassin.apache.org> **SpamAssassin**) suite. Different to the generic filter interface S-nail will automatically add the correct arguments for a given command and has the necessary knowledge to parse the program's output. A default value for `spamc-command[427]` will have been compiled into the S-nail binary if `spamc(1)` has been found in `PATH[475]` during compilation. Shall it be necessary to define a specific connection type (rather than using a configuration file for that), the variable `spamc-arguments[428]` can be used as in, e.g., `-d server.example.com -p 783`. It is also possible to specify a per-user configuration via `spamc-user[429]`. Note that this interface doesn't inspect the `is-spam` flag of a message for the command `spamforget[182]`.

`spamd` S-nail will directly communicate with the `spamassassin(1)` daemon `spamd(1)` via a path-based `unix(4)` stream socket as specified in `spamd-socket[430]`. It is possible to specify a per-user configuration via `spamd-user[431]`.

filter generic spam filter support via freely configurable hooks. This interface is ment for programs like *bogofilter(1)* and *sylfilter(1)* and requires according behaviour in respect to the hooks' exit status for at least the command **spamrate**[184] ('0' meaning a message is spam, '1' for non-spam, '2' for unsure and any other return value indicating a hard error); since the hooks can include shell code snippets diverting behaviour can be intercepted as necessary. The hooks *arespamfilter -ham*[432], *spamfilter-noham*[433], *spamfilter-nospam*[434], *spamfilter-rate*[435] and *spamfilter-spam*[436]; the manual section **Handling spam**[37] contains examples for some programs. The process environment of the hooks will have the variables *NAIL_TMPDIR*[391], *TMPCMDIR*[479] and *NAIL_FILENAME_GENERATED*[393] set. Note that spam score support for **spamrate**[184] isn't supported unless the [Option]tional regular expression support is available and the *spamfilter-rate-scanscore*[437] variable is set.

spam-maxsize

[Option] Messages that exceed this size won't be passed through to the configured *spam-interface*[425]. The default is 420000 bytes.

spamc-command

[Option] The path to the *spamc(1)* program for the *spamc spam-interface*[425]. Note that the path is not expanded, but used "as is". A fallback path will have been compiled into the S-nail binary if the executable had been found during compilation.

spamc-arguments

[Option] Even though S-nail deals with most arguments for the *spamc spam-interface*[425] automatically, it may at least sometimes be desirable to specify connection-related ones via this variable, e.g., `-d server.example.com -p 783`.

spamc-user

[Option] Specify a username for per-user configuration files for the *spamc spam-interface*[425]. If this is set to the empty string then S-nail will use the name of the current *user*[454].

spamd-socket

[Option] Specify the path of the *unix(4)* domain socket on which *spamd(1)* listens for connections for the *spamd spam-interface*[425]. Note that the path is not expanded, but used "as is".

spamd-user

[Option] Specify a username for per-user configuration files for the *spamd spam-interface*[425]. If this is set to the empty string then S-nail will use the name of the current *user*[454].

spamfilter-ham, spamfilter-noham, spamfilter-nospam, spamfilter-rate, spamfilter-spam

[Option] Command and argument hooks for the *filter spam-interface*[425]. The manual section **Handling spam**[37] contains examples for some programs.

spamfilter-rate-scanscore

[Option] Because of the generic nature of the *filter spam-interface*[425] spam scores are not supported for it by default, but if the [Option]tional regular expression support is available then setting this variable can be used to overcome this restriction. It is interpreted as follows: first a number (digits) is parsed that must be followed by a semicolon ';' and a regular expression. Then the latter is used to parse the first output line of the *spamfilter-rate*[435] hook, and, in case the evaluation is successful, the group that has been specified via the number is interpreted as a floating point scan score.

ssl-ca-dir

[Option] Specifies a directory with CA certificates in PEM (Pricacy Enhanced Mail) for verification of of SSL/TLS server certificates. See *SSL_CTX_load_verify_locations(3)* for more information.

ssl-ca-file [Option] Specifies a file with CA certificates in PEM format for verification of SSL/TLS server certificates. See `SSL_CTX_load_verify_locations(3)` for more information.

ssl-cert-USER@HOST, ssl-cert-HOST, ssl-cert

[Option] Variable chain that sets the file name for a SSL/TLS client certificate required by some servers. This is a direct interface to the `Certificate` slot of the `SSL_CONF_cmd(3)` function of the OpenSSL library, if available.

ssl-cipher-list-USER@HOST, ssl-cipher-list-HOST, ssl-cipher-list

[Option] Specifies a list of ciphers for SSL/TLS connections. This is a direct interface to the `CipherString` slot of the `SSL_CONF_cmd(3)` function of the OpenSSL library, if available; see `ciphers(1)` for more information. By default S-nail doesn't set a list of ciphers, which in effect will use a *ssl-protocol*[447] specific cipher (protocol standards ship with a list of acceptable ciphers), possibly cramped to what the actually used SSL/TLS library supports – the manual section **An example configuration**[34] also contains a SSL/TLS use case.

ssl-config-file

[Option] If this variable is set S-nail will call `CONF_modules_load_file(3)` to allow OpenSSL to be configured according to the host system wide security settings. If a non-empty value is given then this will be used to specify the configuration file to be used instead of the global OpenSSL default; note that in this case it is an error if the file cannot be loaded. The application name will always be passed as “s-nail”.

ssl-crl-file [Option] Specifies a file that contains a CRL in PEM format to use when verifying SSL/TLS server certificates.

ssl-crl-dir [Option] Specifies a directory that contains files with CRLs in PEM format to use when verifying SSL/TLS server certificates.

ssl-key-USER@HOST, ssl-key-HOST, ssl-key

[Option] Variable chain that sets the file name for the private key of a SSL/TLS client certificate. If unset, the name of the certificate file is used. The file is expected to be in PEM format. This is a direct interface to the `PrivateKey` slot of the `SSL_CONF_cmd(3)` function of the OpenSSL library, if available.

ssl-method-USER@HOST, ssl-method-HOST, ssl-method

[Option] [Obsolete] Please use the newer and more flexible *ssl-protocol*[447] instead: if both values are set, *ssl-protocol*[447] will take precedence! Can be set to the following values, the actually used *ssl-protocol*[447] specification to which it is mapped is shown in parenthesis: `tlsv1.2` (-ALL, `TLsv1.2`), `tlsv1.1` (-ALL, `TLsv1.1`), `tlsv1` (-ALL, `TLsv1`) and `ssl3` (-ALL, `SSLv3`); the special value `auto` is mapped to `ALL`, `-SSLv2` and thus includes the `SSLv3` protocol. Note that `SSLv2` is no longer supported at all.

ssl-protocol-USER@HOST, ssl-protocol-HOST, ssl-protocol

[Option] Specify the used SSL/TLS protocol. This is a direct interface to the `Protocol` slot of the `SSL_CONF_cmd(3)` function of the OpenSSL library, if available; otherwise an S-nail internal parser is used which understands the following subset of (case-insensitive) command strings: `SSLv3`, `TLsv1`, `TLsv1.1` and `TLsv1.2`, as well as the special value `ALL`. Multiple specifications may be given via a comma-separated list which ignores any whitespace. An optional '+' plus prefix will enable a protocol, a '-' minus prefix will disable it, so that `-ALL`, `TLsv1.2` will enable only the `TLsv1.2` protocol.

It depends upon the used TLS/SSL library which protocols are actually supported and which protocols are used if *ssl-protocol*[447] is not set, but note that `SSLv2` is no longer supported at all and actively disabled. Especially for older protocols explicitly securing *ssl-cipher-list*[441] may be worthwhile, see **An example configuration**[34].

ssl-rand-egd

[Option] Gives the pathname to an entropy daemon socket, see `RAND_egd(3)`. Not all SSL/TLS libraries support this.

ssl-rand-file

[Option] Gives the pathname to a file with entropy data, see `RAND_load_file(3)`. If the file is a regular file writable by the invoking user, new data is written to it after it has been loaded.

ssl-verify-USER@HOST, ssl-verify-HOST, ssl-verify

[Option] Variable chain that sets the action to be performed if an error occurs during SSL/TLS server certificate validation. Valid (case-insensitive) values are `strict` (fail and close connection immediately), `ask` (ask whether to continue on standard input), `warn` (print a warning and continue), `ignore` (do not perform validation). The default is `ask`.

stealthmua

If only set without an assigned value, then this option inhibits the generation of the `Message-ID:` and `User-Agent:` header fields that include obvious references to S-nail. There are two pitfalls associated with this: First, the message id of outgoing messages is not known anymore. Second, an expert may still use the remaining information in the header to track down the originating mail user agent. If set to the value `noagent`, then the mentioned `Message-ID:` suppression doesn't occur.

toplines

If defined, gives the number of lines of a message to be printed out with the top command; normally, the first five lines are printed.

ttycharset

The character set of the terminal S-nail operates on, and the one and only supported character set that S-nail can use if no character set conversion capabilities have been compiled into it, in which case it defaults to ISO-8859-1 unless it can deduce a value from the `LC_CTYPE[466]` locale environment. Refer to the section **Character sets [23]** for the complete picture about character sets.

user-HOST, user

[v15-compat] Variable chain that sets a global fallback user name, which is used in case none has been given in the protocol and account-specific URL. This variable defaults to the value of `USER[480]`.

version, version-major, version-minor, version-update

S-nail version information: the first variable contains a string containing the complete version identification – this is identical to the output of the command `version`. The latter three contain only digits: the major, minor and update version numbers.

ENVIRONMENT[8]

The term “environment variable” should be considered an indication that the following variables are either standardized as being vivid parts of process environments, or are commonly found in there. Unless otherwise explicitly noted they integrate into the normal variable handling, as documented above, from S-nails point of view.

COLUMNS

The user's preferred width in column positions for the terminal screen or window. Queried and used once on program startup.

DEAD

The name of the file to use for saving aborted messages. This defaults to `dead.letter` in the user's `HOME[462]` directory.

EDITOR

Pathname of the text editor to use in the `edit[98]` command and `~e[240]` **TILDE ESCAPES[6]**. A default editor is used if this value is not defined.

- HOME** The user's home directory. This variable is only used when it resides in the process environment. Use **setenv**[174] to update the value at runtime.
- LANG, LC_ALL, LC_COLLATE, LC_CTYPE, LC_MESSAGES**
See **locale(7)** and **Character sets**[23].
- LINES** The user's preferred number of lines on a page or the vertical screen or window size in lines. Queried and used once on program startup.
- LISTER** Pathname of the directory lister to use in the **folders**[108] command when operating on local mailboxes. Default is **ls(1)** (path search through **SHELL**[476]).
- MBOX** The name of the user's mbox file. Supports a logical subset of the special conventions that are documented for the **file**[105] command and the **folder**[367] option. The fallback default is **mbox** in the user's **HOME**[462] directory.
- MAILRC** Is used as a startup file instead of **~/mailrc**[68] if set. When S-nail scripts are invoked on behalf of other users, this variable should be set to **/dev/null** to avoid side-effects from reading their configuration files. This variable is only used when it resides in the process environment.
- NAIL_NO_SYSTEM_RC**
If this variable is set then reading of **/usr/local/etc/s-nail.rc**[67] at startup is inhibited, i.e., the same effect is achieved as if S-nail had been started up with the option **-n**[53]. This variable is only used when it resides in the process environment.
- NETRC** [v15-compat] [Option] This variable overrides the default location of the user's **.netrc** file.
- PAGER** Pathname of the program to use in the more command or when the **crt**[359] variable is set. The default paginator is **more(1)** (path search through **SHELL**[476]).
- PATH** A list of directories that is searched by the shell when looking for commands (as such only recognized in the process environment).
- SHELL** The shell to use for the commands **!**[74], **shell**[175], the **~!**[228] **TILDE ESCAPES**[6] and when starting subprocesses. A default shell is used if this option is not defined.
- SYSV3** Changes the letters printed in the first column of a header summary.
- TERM** [Option] The terminal type for which output is to be prepared.
- TMPDIR** Used as directory for temporary files instead of **/tmp**, if set. This variable is only used when it resides in the process environment. Use **setenv**[174] to update the value at runtime.
- USER** Force identification as the given user, i.e., identical to the **-u**[60] command line option. This variable is only used when it resides in the process environment. Use **setenv**[174] to update the value at runtime, but note that doing so won't trigger any of those validation checks that were performed on program startup (again).
- VISUAL** Pathname of the text editor to use in the **visual**[222] command and **~v**[256] **TILDE ESCAPES**[6].

FILES[9]

- ~/mailrc**[68] File giving initial commands.
- /usr/local/etc/s-nail.rc**[67]
System wide initialization file.
- ~/mime.types** Personal MIME types, see **The mime.types files**[32].

`/etc/mime.types` System wide MIME types, see **The mime.types files**[32].

`~/.netrc` [v15-compat] [Option] The default location of the users `.netrc` file – the section **The .netrc file**[33] documents the file format.

The mime.types files[32]

When sending messages S-nail tries to determine the content type of all attachments. When displaying message content or attachments S-nail uses the content type to decide whether it can directly display data or whether it needs to deal with content handlers, as can be defined via *pipe-TYPE/SUBTYPE*[390] (and *pipe-EXTENSION*[396]) variables, to do so.

It learns about MIME types and how to treat them by reading `mime.types` files, the loading of which can be controlled by setting the variable *mimetypes-load-control*[380]. (The command **mimetype** [133] can also be used to deal with MIME types.) `mime.types` files have the following syntax:

```
type/subtype extension [extension ...]
```

where `type/subtype` are strings describing the file contents, and one or multiple extensions, separated by whitespace, name the part of a filename starting after the last dot (of interest). Comments may be introduced anywhere on a line with a number sign '#', causing the remaining line to be discarded. S-nail also supports an extended, non-portable syntax in specially crafted files, which can be loaded via the alternative value syntax of *mimetypes-load-control*[380] and prepends an optional `type-marker`:

```
[type-marker ]type/subtype extension [extension ...]
```

The following type markers are supported:

- @ Treat message parts with this content as plain text.
- @t@ The same as plain @.
- @h@ Treat message parts with this content as HTML tagsoup. If the [Option]al HTML-tagsoup-to-text converter is not available treat the content as plain text instead.
- @H@ Likewise @h@ but instead of falling back to plain text require an explicit content handler to be defined.

Further reading: for sending messages: *mime-allow-text-controls*[308]. For reading etc. messages: **Viewing HTML mail and MIME attachments**[20], *mime-counter-evidence*[379], *pipe-TYPE/SUBTYPE*[390], *pipe-EXTENSION*[396].

The .netrc file[33]

The `.netrc` file contains user credentials for machine accounts. The default location in the user's HOME [462] directory may be overridden by the `NETRC` [473] environment variable. The file consists of space, tabulator or newline separated tokens. S-nail implements a parser that supports a superset of the original BSD syntax, but users should nonetheless be aware of portability glitches of that file format, shall their `.netrc` be usable across multiple programs and platforms:

- BSD doesn't support single, but only double quotation marks, e.g., `password="pass with spaces"`.
- BSD (only?) supports escaping of single characters via a backslash (e.g., a space can be escaped via `'\ '`), in- as well as outside of a quoted string.
- BSD doesn't require the final quotation mark of the final user input token.
- At least Hewlett-Packard seems to support a format which also allows tokens to be separated with commas – this format is not supported!
- Whereas other programs may require that the `.netrc` file is accessible by only the user if it contains a **password** token for any other **login** than anonymous, S-nail will always require these strict permissions.

Of the following list of supported tokens S-nail only uses (and caches) **machine**, **login** and **password**:

machine *name*

The hostname of the entries' machine, lowercase-normalized by S-nail before use. Any further file content, until either end-of-file or the occurrence of another **machine** or a **default** first-class token is bound (only related) to the machine *name*.

As an extension that shouldn't be the cause of any worries S-nail supports a single wildcard prefix for *name*:

```
machine *.example.com login USER password PASS
machine pop3.example.com login USER password PASS
machine smtp.example.com login USER password PASS
```

which would match *xy.example.com* as well as *pop3.example.com*, but neither *example.com* nor *local.smtp.example.com*. Note that in the example neither *pop3.example.com* nor *smtp.example.com* will be matched by the wildcard, since the exact matches take precedence (it is however faster to specify it the other way around).

default This is the same as **machine** except that it is a fallback entry that is used shall none of the specified machines match; only one default token may be specified, and it must be the last first-class token.

login *name*

The user name on the remote machine.

password *string*

The user's password on the remote machine.

account *string*

Supply an additional account password. This is merely for FTP purposes.

macdef *name*

Define a macro. A macro is defined with the specified *name*; it is formed from all lines beginning with the next line and continuing until a blank line is (consecutive newline characters are) encountered. (Note that **macdef** entries cannot be utilized by multiple machines, too, but must be defined following the **machine** they are intended to be used with.) If a macro named *init* exists, it is automatically run as the last step of the login process. This is merely for FTP purposes.

EXAMPLES[10]

An example configuration[34]

```
# This example assumes v15.0 compatibility mode
set v15-compat

# Where are the up-to-date SSL certificates?
#set ssl-ca-dir=/etc/ssl/certs
set ssl-ca-file=/etc/ssl/certs/ca-certificates.crt

# (Since we manage up-to-date ones explicitly, don't use any,
# possibly outdated, default certificates shipped with OpenSSL
set ssl-no-default-ca

# Don't use protocols older than TLS v1.2.
# Change this only when the remote server doesn't support it:
# maybe use ssl-protocol-HOST (or -USER@HOST) syntax to define
```

```

# such explicit exceptions, then
set ssl-protocol="-ALL,+TLSv1.2"

# Explicitly define the list of ciphers, which may improve security,
# especially with protocols older than TLS v1.2. See ciphers(1).
# Hint: it is important to include "@STRENGTH": only with it the
# final list will be sorted by algorithm strength.
# This is an example: in reality it is possibly best to only use
# ssl-cipher-list-HOST (or -USER@HOST), as necessary, again..
set ssl-cipher-list="ALL:!aNULL:!MEDIUM:!LOW:\
    !MD5:!RC4:!EXPORT:@STRENGTH"

# Request strict transport security checks!
set ssl-verify=strict

# Essential setting: select allowed character sets
set sendcharsets=utf-8,iso-8859-1

# When sending messages, wait until the Mail-Transfer-Agent finishes.
# Only like this you'll be able to see errors reported through the
# exit status of the MTA (including the builtin SMTP one)!
set sendwait

# Only use builtin MIME types, no mime.types(5) files
set mimetypes-load-control

# Default directory where we act in (relative to $HOME)
set folder=mail MBOX=+mbox.mbox record=+sent.mbox \
    DEAD=+dead.mbox

# Make "file mymbox" and "file myrec" go to..
shortcut mymbox %:+mbox.mbox myrec +sent.mbox

# Not really optional, e.g., for S/MIME
set from="Your Name <youremail@domain>"

# It may be necessary to set hostname and/or smtp-hostname
# if the "SERVER" of smtp and "domain" of from don't match.
# The 'urlencode' command can be used to encode USER and PASS
set smtp=(smtp[s]/submission)://[USER[:PASS]@]SERVER[:PORT] \
    smtp-auth=login/plain... \
    smtp-use-starttls

# Never refuse to start into interactive mode, and more
set emptystart \
    colour-pager crt= followup-to followup-to-honour=ask-yes \
    history-gabby mime-counter-evidence=6 \
    prompt=" [31m?[\$ \@]\& [0m" \
    NAIL_HISTFILE=+.s-nailhist NAIL_HISTSIZE=-1 \
    reply-to-honour=ask-yes

# When 'p'rinting messages, show only these headers

```

```

# (use `P`rint for all headers and `S`how for raw message)
retain date from to cc subject

# Some mailing lists
mlist @xyz-editor.xyz$ @xyzf.xyz$
mlsubscribe ^xfans@xfans.xyz$

# A real life example of a very huge free mail provider
account XoogIX {
    localopts yes
    # (The plain smtp:// proto is optional)
    set smtp=USER:PASS@smtp.gmXil.com smtp-use-starttls
    set from="Your Name <youremail@domain>"
}

# Here is a pretty large one which does not allow sending mails
# if there is a domain name mismatch on the SMTP protocol level,
# which would bite us if the value of from does not match, e.g.,
# for people who have a sXXXXeforge project and want to speak
# with the mailing list under their project account (in from),
# still sending the message through their normal mail provider
account XandEX {
    localopts true
    set smtp=smtps://USER:PASS@smtp.yaXXex.ru:465 \
        hostname=yaXXex.com smtp-hostname=
    set from="Your Name <youremail@domain>"
}

# Create some new commands so that, e.g., `ls /tmp` will..
ghost l      !ls -aFtr
ghost L      !ls -aFt
ghost ll     !ls -aFltr
ghost Ll     !ls -aFlt
ghost la     !ls -aFr
ghost La     !ls -aF
ghost lla    !ls -aFlr
ghost Lla    !ls -aFl
ghost lS     !ls -aFrS
ghost LS     !ls -aFS
ghost llS    !ls -aFlrS
ghost LlS    !ls -aFlS

```

When storing passwords in `~/.mailrc` [68] appropriate permissions should be set on this file with `$ chmod 0600 ~/.mailrc`. If the [Option] `alnetrc-lookup` [309] is available user credentials can be stored in the central `.netrc` file instead; e.g., here is a different version of the example account that sets up SMTP and POP3:

```

account XandEX {
    localopts true
    set netrc-lookup
    #set agent-shell-lookup="gpg -d .pass.gpg"

    set smtp=smtps://smtp.yXXXXXx.ru:465 \

```

```

smtp-hostname= hostname=yXXXXx.com
set pop3-keepalive=240 pop3-no-apop-pop.yXXXXx.ru
ghost xp fi pop3s://pop.yXXXXx.ru
}

```

and, in the `.netrc` file:

```
machine *.yXXXXx.ru login USER password PASS
```

If the also [Option]al *agent-shell-lookup*[343] is available things could be diversified further by using encrypted password storage: for this, don't specify `password PASS` in the `.netrc` file and instead uncomment the line that defines agent lookup in the example **account**[76] above, then create the encrypted password storage file `.pass.gpg`:

```

$ echo PASS > .pass
$ gpg -e .pass
$ eval `gpg-agent --daemon \
      --pinentry-program=/usr/bin/pinentry-curses \
      --max-cache-ttl 99999 --default-cache-ttl 99999`

```

This configuration should now work just fine (use the `-d` [43] command line option for a(n almost) dry-run):

```
$ echo text | s-nail -vv -AXandex -s Subject some@where
```

Signed and encrypted messages with S/MIME[35]

[Option] S/MIME provides two central mechanisms: message signing and message encryption. A signed message contains some data in addition to the regular text. The data can be used to verify that the message was sent using a valid certificate, that the sender's address in the message header matches that in the certificate, and that the message text has not been altered. Signing a message does not change its regular text; it can be read regardless of whether the recipient's software is able to handle S/MIME.

It is thus usually possible to sign all outgoing messages if so desired. Encryption, in contrast, makes the message text invisible for all people except those who have access to the secret decryption key. To encrypt a message, the specific recipient's public encryption key must be known. It is thus not possible to send encrypted mail to people unless their key has been retrieved from either previous communication or public key directories. A message should always be signed before it is encrypted. Otherwise, it is still possible that the encrypted message text is altered.

A central concept to S/MIME is that of the certification authority (CA). A CA is a trusted institution that issues certificates. For each of these certificates it can be verified that it really originates from the CA, provided that the CA's own certificate is previously known. A set of CA certificates is usually delivered with OpenSSL and installed on your system. If you trust the source of your OpenSSL software installation, this offers reasonable security for S/MIME on the Internet. In general, a certificate cannot be more secure than the method its CA certificate has been retrieved with, though. Thus if you download a CA certificate from the Internet, you can only trust the messages you verify using that certificate as much as you trust the download process.

The first thing you need for participating in S/MIME message exchange is your personal certificate, including a private key. The certificate contains public information, in particular your name and your email address, and the public key that is used by others to encrypt messages for you, and to verify signed messages they supposedly received from you. The certificate is included in each signed message you send. The private key must be kept secret. It is used to decrypt messages that were previously encrypted with your public key, and to sign messages.

For personal use it is recommended that you get a S/MIME certificate from one of the major CAs on the Internet using your WWW browser. (Many CAs offer such certificates for free.) You will usually receive a combined certificate and private key in PKCS#12 format which S-nail does not directly accept. To convert it

to PEM format, use the following shell command:

```
$ openssl pkcs12 -in cert.p12 -out cert.pem -clcerts -nodes
```

If you omit the `-nodes` parameter, you can specify an additional “PEM pass phrase” for protecting the private key. S-nail will then ask you for that pass phrase each time it signs or decrypts a message. Afterwards you can then use

```
set smime-sign-cert-myname@myisp.example=cert.pem
```

to make this private key and certificate known to S-nail. You can now sign outgoing messages, just set

```
set smime-sign
```

From each signed message you send, the recipient can fetch your certificate and use it to send encrypted mail back to you. Accordingly if somebody sends you a signed message, you can do the same. First use the **verify**[221] command to check the validity of the certificate. After that, retrieve the certificate and tell S-nail that it should use it for encryption:

```
certsave filename
set smime-encrypt-USER@HOST=filename
```

You should carefully consider if you prefer to store encrypted messages in decrypted form. If you do, anybody who has access to your mail folders can read them, but if you do not, you might be unable to read them yourself later if you happen to lose your private key. The **decrypt** [90] command saves messages in decrypted form, while the **save** [168], **copy** [87], and **move** [136] commands leave them encrypted.

Note that neither S/MIME signing nor encryption applies to message subjects or other header fields. Thus they may not contain sensitive information for encrypted messages, and cannot be trusted even if the message content has been verified. When sending signed messages, it is recommended to repeat any important header information in the message text.

Using CRLs with S/MIME or SSL/TLS[36]

[Option] Certification authorities (CAs) issue certificate revocation lists (CRLs) on a regular basis. These lists contain the serial numbers of certificates that have been declared invalid after they have been issued. Such usually happens because the private key for the certificate has been compromised, because the owner of the certificate has left the organization that is mentioned in the certificate, etc. To seriously use S/MIME or SSL/TLS verification, an up-to-date CRL is required for each trusted CA. There is otherwise no method to distinguish between valid and invalidated certificates. S-nail currently offers no mechanism to fetch CRLs, nor to access them on the Internet, so you have to retrieve them by some external mechanism.

S-nail accepts CRLs in PEM format only; CRLs in DER format must be converted, like, e. g.:

```
$ openssl crl -inform DER -in crl.der -out crl.pem
```

To tell S-nail about the CRLs, a directory that contains all CRL files (and no other files) must be created. The `smime-crl-dir`[418] or `ssl-crl-dir`[444] variables, respectively, must then be set to point to that directory. After that, S-nail requires a CRL to be present for each CA that is used to verify a certificate.

Handling spam[37]

[Option] S-nail can make use of several spam interfaces for the purpose of identification of, and, in general, dealing with spam messages. A precondition of most commands in order to function is that the `spam-interface`[425] variable is set to one of the supported interfaces. Once messages have been identified as spam their (volatile) `is-spam` state can be prompted: the `:s` and `:S` message specifications will address respective messages and their `attrlist`[344] entries will be used when displaying the `headline`[373] in the header display.

- **spamrate**[184] rates the given messages and sets their `is-spam` flag accordingly. If the spam interface offers spam scores those can also be displayed in the header display by including the ‘%\$’ format in the `headline`[373] variable.
- **spamham**[183], **spamspam**[186] and **spamforget**[182] will interact with the Bayesian filter of the chosen interface and learn the given messages as “ham” or “spam”, respectively; the last command can be used to cause “unlearning” of messages; it adheres to their current `is-spam` state and thus reverts previous teachings.
- **spamclear**[181] and **spamset**[185] will simply set and clear, respectively, the mentioned volatile `is-spam` message flag, without any interaction with an interface.

The `spamassassin(1)` based `spam-interface`[425]s `spamd` and `spamd` require a running instance of the `spamd(1)` server in order to function, started with the option `--allow-tell` shall Bayesian filter learning be possible. `spamd` only works via a local path-based `unix(4)` socket, but otherwise the following will be equivalently fine:

```
$ spamd -i localhost:2142 -i /tmp/.spamsock -d [-L] [-l]
$ spamd --listen=localhost:2142 --listen=/tmp/.spamsock \
  --daemonize [--local] [--allow-tell]
```

Thereafter S-nail can make use of these interfaces:

```
$ s-nail -Sspam-interface=spamd -Sspam-maxsize=500000 \
  -Sspamd-socket=/tmp/.spamsock -Sspamd-user=

$ s-nail -Sspam-interface=spamc -Sspam-maxsize=500000 \
  -Sspamc-command=/usr/local/bin/spamc \
  -Sspamc-arguments="-U /tmp/.spamsock" -Sspamc-user=
or
$ s-nail -Sspam-interface=spamc -Sspam-maxsize=500000 \
  -Sspamc-command=/usr/local/bin/spamc \
  -Sspamc-arguments="-d localhost -p 2142" -Sspamc-user=
```

Using the generic filter approach allows usage of programs like `bogofilter(1)` and `sylfilter(1)`. Here is an example for the former, requiring it to be accessible via `PATH`[475]:

```
$ s-nail -Sspam-interface=filter -Sspam-maxsize=500000 \
  -Sspamfilter-ham="bogofilter -n" \
  -Sspamfilter-noham="bogofilter -N" \
  -Sspamfilter-nospam="bogofilter -S" \
  -Sspamfilter-rate="bogofilter -TTu 2>/dev/null" \
  -Sspamfilter-spam="bogofilter -s" \
  -Sspamfilter-rate-scanscore="1;^(.+)$"
```

Because messages must exist on local storage in order to be scored (or used for Bayesian filter training), it is possibly a good idea to perform the local spam check last:

```
define spandelhook {
  # Server side DCC
  spamset (header x-dcc-brand-metrics "bulk")
  # Server-side spamassassin(1)
  spamset (header x-spam-flag "YES")
  del :s # TODO we HAVE to be able to do `spamrate :u ! :sS`
  move :S +maybe-spam
  spamrate :u
  del :s
}
```

```

    move :S +maybe-spam
}
set folder-hook-FOLDER=spandelhook

```

See also the documentation for the variables *spam-interface*[425], *spam-maxsize*[426], *spamc-command*[427], *spamc-arguments*[428], *spamc-user*[429], *spamd-socket*[430], *spamd-user*[431], *spamfilter-ham*[432], *spamfilter-noham*[433], *spamfilter-nospam*[434], *spamfilter-rate*[435] and *spamfilter-rate-scanscore*[437].

SEE ALSO[11]

bogofilter(1), bzip2(1), file(1), fmt(1), gpg(1), gpg-agent(1), gzip(1), less(1), more(1), newaliases(1), openssl(1), postfix(1), printf(1), sendmail(1), sh(1), spamassassin(1), spamc(1), spamd(1), sylfilter(1), vacation(1), xterm(1), xz(1), editline(3), iconv(3), readline(3), setlocale(3), ssl(3), aliases(5), locale(7), mailaddr(7), re_format(7), exim(8), mailwrapper(8), sendmail(8)

HISTORY[12]

A *mail* command appeared in Version 1 AT&T Unix. Berkeley Mail was written in 1978 by Kurt Shoens. This man page is derived from from “The Mail Reference Manual” originally written by Kurt Shoens. “Heirloom Mailx” enhancements are maintained and documented by Gunnar Ritter. “S-nail” is maintained and documented by Steffen (Daode) Nurpmeso.

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1, 2003 Edition, Standard for Information Technology – Operating System Interface (POSIX), The Open Group Base Specifications Issue 6, Copyright © 2001-2003 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at *OpenGroup.org*:

<http://www.opengroup.org/unix/online.html>

Redistribution of this material is permitted so long as this notice remains intact.

AUTHORS[13]

Kurt Shoens,
 Christos Zoulas,
 Gunnar Ritter,
 Steffen Nurpmeso <s-nail-users@lists.sourceforge.net>

CAVEATS[14]

The character set conversion uses and relies upon the *iconv*(3) function. Its functionality differs widely between the various system environments S-nail runs on.

Limitations with POP3 mailboxes are: It is not possible to edit messages, they can only be copied and deleted. The line count for the header display is only appropriate if the entire message has been downloaded from the server. The status field of a message is maintained by the server between connections; some servers do not update it at all, and with a server that does, the **exit**[102] command will not cause the message status to be reset. The **newmail**[141] command and the *newmail*[387] variable have no effect. It is not possible to rename or to remove POP3 mailboxes.

If a “RUBOUT” (interrupt,control-C) is typed while a POP3 operation is in progress, S-nail will wait until the operation can be safely aborted, and will then return to the command loop and print the prompt again. When a second “RUBOUT” is typed while S-nail is waiting for the operation to complete, the operation itself will be cancelled. In this case, data that has not been fetched yet will have to be fetched before the next command can be performed. If the cancelled operation was using an SSL/TLS encrypted channel, an error in the SSL transport will very likely result and render the connection unusable.

As S-nail is a mail user agent, it provides only basic SMTP services. If it fails to contact its upstream SMTP server, it will not make further attempts to transfer the message at a later time, and it does not leave other information about this condition than an error message on the terminal and an entry in DEAD[460]. This is usually not a problem if the SMTP server is located in the same local network as the computer on which S-nail is run. However, care should be taken when using a remote server of an ISP; it might be better to set up a local SMTP server then which just acts as a proxy.

S-nail immediately contacts the SMTP server (or `sendmail(1)`). It would not make much sense for S-nail to defer outgoing mail since SMTP servers usually provide much more elaborated delay handling than S-nail could perform as a client.

BUGS[15]

See the file TODO from the distribution or the repository.