

## NAME

muttrc – Configuration file for the Mutt Mail User Agent

## DESCRIPTION

A mutt configuration file consists of a series of “commands”. Each line of the file may contain one or more commands. When multiple commands are used, they must be separated by a semicolon (“;”).

The hash mark, or pound sign (“#”), is used as a “comment” character. You can use it to annotate your initialization file. All text after the comment character to the end of the line is ignored.

Single quotes (“’”) and double quotes (“”) can be used to quote strings which contain spaces or other special characters. The difference between the two types of quotes is similar to that of many popular shell programs, namely that a single quote is used to specify a literal string (one that is not interpreted for shell variables or quoting with a backslash [see next paragraph]), while double quotes indicate a string which should be evaluated. For example, backticks are evaluated inside of double quotes, but not single quotes.

\ quotes the next character, just as in shells such as bash and zsh. For example, if want to put quotes (“”) inside of a string, you can use “\” to force the next character to be a literal instead of interpreted character.

“\” means to insert a literal “\” into the line. “\n” and “\r” have their usual C meanings of linefeed and carriage-return, respectively.

A “\” at the end of a line can be used to split commands over multiple lines, provided that the split points don’t appear in the middle of command names.

It is also possible to substitute the output of a Unix command in an initialization file. This is accomplished by enclosing the command in backticks (*command*).

UNIX environment variables can be accessed like the way it is done in shells like sh and bash: Prepend the name of the variable by a dollar (“\$”) sign.

## COMMANDS

**alias** [-group *name* [...]] *key* *address* [, *address* [ ... ]]

**unalias** [ \* | *key* ]

**alias** defines an alias *key* for the given addresses. Each *address* will be resolved into either an email address (user@example.com) or a named email address (User Name <user@example.com>). The address may be specified in either format, or in the format “user@example.com (User Name)”. **unalias** removes the alias corresponding to the given *key* or all aliases when “\*” is used as an argument. The optional **-group** argument to **alias** causes the aliased address(es) to be added to the named *group*.

**group** [-group *name*] [-rx *EXPR* [ ... ]] [-addr *address* [ ... ]]

**ungroup** [-group *name*] [ \* | [[-rx *EXPR* [ ... ]] [-addr *address* [ ... ]]]

**group** is used to directly add either addresses or regular expressions to the specified group or groups. The different categories of arguments to the **group** command can be in any order. The flags **-rx** and **-addr** specify what the following strings (that cannot begin with a hyphen) should be interpreted as: either a regular expression or an email address, respectively. **ungroup** is used to remove addresses or regular expressions from the specified group or groups. The syntax is similar to the **group** command, however the special character \* can be used to empty a group of all of its contents.

These address groups can also be created implicitly by the **alias**, **lists**, **subscribe** and **alternates** commands by specifying the optional **-group** option.

Once defined, these address groups can be used in patterns to search for and limit the display to messages matching a group.

**alternates** [-group *name*] *regex* [ *regex* [ ... ]]

**unalternates** [ \* | *regex* [ *regex* [ ... ]]

**alternates** is used to inform mutt about alternate addresses where you receive mail; you can use regular expressions to specify alternate addresses. This affects mutt’s idea about messages from

you, and messages addressed to you. **unalter nates** removes a regular expression from the list of known alternates. The **-group** flag causes all of the subsequent regular expressions to be added to the named group.

**alternative\_order** *type[/subtype]* [ ... ]

**unalternative\_order** [ \* | *type/subtype* ] [...]

**alternative\_order** command permits you to define an order of preference which is used by mutt to determine which part of a **multipart/alternative** body to display. A subtype of "\*" matches any subtype, as does an empty subtype. **unalternative\_order** removes entries from the ordered list or deletes the entire list when "\*" is used as an argument.

**attachments** [ + | - ] *disposition mime-type*

**unattachments** [ + | - ] *disposition mime-type*

**attachments ?**

**unattachments \***

**attachments** specifies what kinds of attachments are used for Mutt's attachment counting and searching support.

*disposition* is the attachment's Content-Disposition type - either inline or attachment. You can abbreviate this to I or A.

The first part of a message or multipart group, if inline, is counted separately than other inline parts. Specify root or R for disposition to count these as attachments. If this first part is of type multipart/alternative, note that its top-level inline parts are also counted via root disposition (if \$count\_alternatives is set).

*disposition* is prefixed by either a + symbol or a - symbol. If it's a +, you're saying that you want to allow this disposition and MIME type to qualify. If it's a -, you're saying that this disposition and MIME type is an exception to previous + rules.

*mime-type* is the MIME type of the attachment you want the command to affect. A MIME type is always of the format major/minor, where major describes the broad category of document you're looking at, and minor describes the specific type within that category. The major part of mime-type must be literal text (or the special token \*), but the minor part may be a regular expression. (Therefore, \*/.\* matches any MIME type.)

The MIME types you give to the attachments directive are a kind of pattern. When you use the attachments directive, the patterns you specify are added to a list. When you use unattachments, the pattern is removed from the list. The patterns are not expanded and matched to specific MIME types at this time - they're just text in a list. They're only matched when actually evaluating a message.

**auto\_view** *type[/subtype]* [ ... ]

**unauto\_view** *type[/subtype]* [ ... ]

This commands permits you to specify that mutt should automatically convert the given MIME types to text/plain when displaying messages. For this to work, there must be a **mailcap(5)** entry for the given MIME type with the **copiousoutput** flag set. A subtype of "\*" matches any subtype, as does an empty subtype.

**mime\_lookup** *type[/subtype]* [ ... ]

**unmime\_lookup** *type[/subtype]* [ ... ]

This command permits you to define a list of "data" MIME content types for which mutt will try to determine the actual file type from the file name, and not use a **mailcap(5)** entry given for the original MIME type. For instance, you may add the **application/octet-stream** MIME type to this list.

**bind** *map1,map2,... key function*

This command binds the given *key* for the given *map* or maps to the given *function*. Multiple maps may be specified by separating them with commas (no whitespace is allowed).

Valid maps are: **generic**, **alias**, **attach**, **browser**, **editor**, **index**, **compose**, **pager**, **pgp**, **postpone**, **mix**.

For more information on keys and functions, please consult the Mutt Manual. Note that the function name is to be specified without angle brackets.

**account-hook** *[!]regex command*

This hook is executed whenever you access a remote mailbox. Useful to adjust configuration settings to different IMAP or POP servers.

**charset-hook** *alias charset*

This command defines an alias for a character set. This is useful to properly display messages which are tagged with a character set name not known to mutt.

**iconv-hook** *charset local-charset*

This command defines a system-specific name for a character set. This is useful when your system's **iconv**(3) implementation does not understand MIME character set names (such as **iso-8859-1**), but instead insists on being fed with implementation-specific character set names (such as **8859-1**). In this specific case, you'd put this into your configuration file:

**iconv-hook iso-8859-1 8859-1**

**message-hook** *[!]pattern command*

Before mutt displays (or formats for replying or forwarding) a message which matches the given *pattern* (or, when it is preceded by an exclamation mark, does not match the *pattern*), the given *command* is executed. When multiple **message-hooks** match, they are executed in the order in which they occur in the configuration file.

**folder-hook** *[!]regex command*

When mutt enters a folder which matches *regex* (or, when *regex* is preceded by an exclamation mark, does not match *regex*), the given *command* is executed.

When several **folder-hooks** match a given mail folder, they are executed in the order given in the configuration file.

**macro** *map key sequence [ description ]*

This command binds the given *sequence* of keys to the given *key* in the given *map* or maps. For valid maps, see **bind**. To specify multiple maps, put only a comma between the maps.

**color** *object [ attribute ... ] foreground background [ regex ]*

**color** *index [ attribute ... ] foreground background [ pattern ]*

**color** *compose composeobject [ attribute ... ] foreground background*

**uncolor** *index pattern [ pattern ... ]*

If your terminal supports color, these commands can be used to assign *foreground/background* combinations to certain objects. Valid objects are: **attachment**, **body**, **bold**, **error**, **header**, **hdrdefault**, **index**, **indicator**, **markers**, **message**, **normal**, **prompt**, **quoted**, **quotedN**, **search**, **signature**, **status**, **tilde**, **tree**, **underline**. If the sidebar is enabled the following objects are also valid: **sidebar\_divider**, **sidebar\_flagged**, **sidebar\_highlight**, **sidebar\_indicator**, **sidebar\_new**, **sidebar\_spoolfile**. The **body** and **header** objects allow you to restrict the colorization to a regular expression. The **index** object permits you to select colored messages by pattern.

Valid composeobjects include **header**, **security\_encrypt**, **security\_sign**, **security\_both**, **security\_none**.

Valid colors include: **white**, **black**, **green**, **magenta**, **blue**, **cyan**, **yellow**, **red**, **default**, **colorN**.

Valid attributes include: **none**, **bold**, **underline**, **reverse**, and **standout**.

**mono** *object attribute [ regex ]*

**mono** *index attribute [ pattern ]*

For terminals which don't support color, you can still assign attributes to objects.

**[un]ignore** *pattern* [ *pattern* ... ]

The **ignore** command permits you to specify header fields which you usually don't wish to see. Any header field whose tag *begins* with an "ignored" pattern will be ignored.

The **unignore** command permits you to define exceptions from the above mentioned list of ignored headers.

**lists** [-group *name*] *regex* [ *regex* ... ]

**unlists** *regex* [ *regex* ... ]

**subscribe** [-group *name*] *regex* [ *regex* ... ]

**unsubscribe** *regex* [ *regex* ... ]

Mutt maintains two lists of mailing list address patterns, a list of subscribed mailing lists, and a list of known mailing lists. All subscribed mailing lists are known. Patterns use regular expressions.

The **lists** command adds a mailing list address to the list of known mailing lists. The **unlists** command removes a mailing list from the lists of known and subscribed mailing lists. The **subscribe** command adds a mailing list to the lists of known and subscribed mailing lists. The **unsubscribe** command removes it from the list of subscribed mailing lists. The **-group** flag adds all of the subsequent regular expressions to the named group.

**mbox-hook** [!] *regex mailbox*

When mutt changes to a mail folder which matches *regex*, *mailbox* will be used as the "mbox" folder, i.e., read messages will be moved to that folder when the mail folder is left.

The first matching **mbox-hook** applies.

**mailboxes** [-notify | -nonotify]

[-poll | -nopoll]

[-label *label*] | -nolabel

*filename*] [ ... ]

**unmailboxes** [ \* | *filename* ... ]

The **mailboxes** specifies folders which can receive mail and which will be checked for new messages. When changing folders, pressing space will cycle through folders with new mail. The **unmailboxes** command is used to remove a file name from the list of folders which can receive mail. If "\*" is specified as the file name, the list is emptied.

**my\_hdr** *string*

**unmy\_hdr** *field*

Using **my\_hdr**, you can define headers which will be added to the messages you compose. **unmy\_hdr** will remove the given user-defined headers.

**hdr\_order** *header1 header2* [ ... ]

With this command, you can specify an order in which mutt will attempt to present headers to you when viewing messages.

**save-hook** [!] *pattern filename*

When a message matches *pattern*, the default file name when saving it will be the given *filename*.

**fcc-hook** [!] *pattern filename*

When an outgoing message matches *pattern*, the default file name for storing a copy (fcc) will be the given *filename*.

**fcc-save-hook** [!] *pattern filename*

This command is an abbreviation for identical **fcc-hook** and **save-hook** commands.

**send-hook** [!] *pattern command*

When composing a message matching *pattern*, *command* is executed. When multiple **send-hooks** match, they are executed in the order in which they occur in the configuration file.

**send2-hook** *[!]pattern command*

Whenever a message matching *pattern* is changed (either by editing it or by using the compose menu), *command* is executed. When multiple **send2-hooks** match, they are executed in the order in which they occur in the configuration file. Possible applications include setting the \$sendmail variable when a message's from header is changed.

**send2-hook** execution is not triggered by use of **enter-command** from the compose menu.

**reply-hook** *[!]pattern command*

When replying to a message matching *pattern*, *command* is executed. When multiple **reply-hooks** match, they are executed in the order in which they occur in the configuration file, but all **reply-hooks** are matched and executed before **send-hooks**, regardless of their order in the configuration file.

**crypt-hook** *regexp key-id*

The crypt-hook command provides a method by which you can specify the ID of the public key to be used when encrypting messages to a certain recipient. The meaning of "key ID" is to be taken broadly: This can be a different e-mail address, a numerical key ID, or even just an arbitrary search string. You may use multiple **crypt-hooks** with the same *regexp*; multiple matching **crypt-hooks** result in the use of multiple *key-ids* for a recipient.

**index-format-hook** *name [!]pattern format-string*

This command is used to inject format strings dynamically into \$index\_format based on pattern matching against the current message.

The \$index\_format expando %@name@ specifies a placeholder for the injection. Index-format-hooks with the same *name* are matched using *pattern* against the current message. Matching is done in the order specified in the .muttrc, with the first match being used. The hook's *format-string* is then substituted and evaluated.

**open-hook** *regexp "command"***close-hook** *regexp "command"***append-hook** *regexp "command"*

These commands provide a way to handle compressed folders. The given **regexp** specifies which folders are taken as compressed (e.g. "\\.\.gz\$"). The commands tell Mutt how to uncompress a folder (**open-hook**), compress a folder (**close-hook**) or append a compressed mail to a compressed folder (**append-hook**). The *command* string is the **printf**(3) like format string, and it should accept two parameters: **%f**, which is replaced with the (compressed) folder name, and **%t** which is replaced with the name of the temporary folder to which to write.

**push** *string***exec** *function [ ... ]*

**push** adds the named *string* to the keyboard buffer. "exec function" is equivalent to "push <function>".

**run** *MuttLisp*

The **run** command evaluates the *MuttLisp* argument. The output of the *MuttLisp* is then executed as a Mutt command, as if it were typed in the muttrc instead.

**score** *pattern value***unscore** *[ \* | pattern ... ]*

The **score** commands adds *value* to a message's score if *pattern* matches it. The **unscore** command removes score entries from the list.

**set** *[no]inv[&|?]  
variable[=value] [ ... ]***toggle** *variable [ ... ]***unset** *variable [ ... ]***reset** *variable [ ... ]*

These commands are used to set and manipulate configuration variables.

Mutt knows four basic types of variables: boolean, number, string and quadoption. Boolean variables can be **set** (true), **unset** (false), or **toggled**. Number variables can be assigned a positive integer value.

String variables consist of any number of printable characters. Strings must be enclosed in quotes if they contain spaces or tabs. You may also use the “C” escape sequences **\n** and **\t** for newline and tab, respectively.

Quadoption variables are used to control whether or not to be prompted for certain actions, or to specify a default action. A value of **yes** will cause the action to be carried out automatically as if you had answered yes to the question. Similarly, a value of **no** will cause the action to be carried out as if you had answered “no.” A value of **ask-yes** will cause a prompt with a default answer of “yes” and **ask-no** will provide a default answer of “no.”

The **reset** command resets all given variables to the compile time defaults. If you reset the special variable **all**, all variables will reset to their compile time defaults.

**setenv** [**?**]*variable* [*value* ]

**unsetenv** *variable*

These alter the environment that Mutt passes on to its child processes. You can also query current environment values by prefixing a “?” character.

**sidebar\_whitelist** *mailbox* [*mailbox* ...]

**unsidebar\_whitelist** [**\*** | *mailbox* ... ]

**sidebar\_whitelist** specifies mailboxes that will always be displayed in the sidebar, even if **\$sidebar\_new\_mail\_only** is set and the mailbox does not contain new mail.

**unsidebar\_whitelist** is used to remove a mailbox from the list of whitelisted mailboxes. Use **unsidebar\_whitelist \*** to remove all mailboxes.

**source** *filename*

The given file will be evaluated as a configuration file.

**spam** *pattern format*

**nospam** *pattern*

These commands define spam-detection patterns from external spam filters, so that mutt can sort, limit, and search on “spam tags” or “spam attributes”, or display them in the index. See the Mutt manual for details.

**subjectrx** *pattern replacement*

**unsubjectrx** [**\*** | *pattern* ]

**subjectrx** specifies a regular expression *pattern* which, if detected in a message subject, causes the subject to be replaced with the *replacement* value. The *replacement* is subject to substitutions in the same way as for the **spam** command: %L for the text to the left of the match, %R for text to the right of the match, and %1 for the first subgroup in the match (etc). If you simply want to erase the match, set it to “%L%R”. Any number of **subjectrx** commands may coexist.

Note this well: the *replacement* value replaces the entire subject, not just the match!

**unsubjectrx** removes a given **subjectrx** from the substitution list. If **\*** is used as the pattern, all substitutions will be removed.

**unhook** [**\*** | *hook-type* ]

This command will remove all hooks of a given type, or all hooks when “\*” is used as an argument. *hook-type* can be any of the **-hook** commands documented above.

**mailto\_allow** *header-field* [ ... ]

**unmailto\_allow** [**\*** | *header-field* ... ]

These commands allow the user to modify the list of allowed header fields in a *mailto:* URL that Mutt will include in the the generated message. By default the list contains only **subject** and **body**, as specified by RFC2368.

#### **echo** *message*

Prints *message* to the message window. After printing the message, echo will pause for the number of seconds specified by \$sleep\_time.

#### **cd** *directory*

Changes the current working directory.

## **PATTERNS**

In various places with mutt, including some of the above mentioned **hook** commands, you can specify patterns to match messages.

### **Constructing Patterns**

A simple pattern consists of a modifier of the form “~*character*”, possibly followed by a parameter against which mutt is supposed to match the object specified by this modifier. For some *characters*, the ~ may be replaced by another character to alter the behavior of the match. These are described in the list of modifiers, below.

With some of these modifiers, the object to be matched consists of several e-mail addresses. In these cases, the object is matched if at least one of these e-mail addresses matches. You can prepend a hat (“^”) character to such a pattern to indicate that *all* addresses must match in order to match the object.

You can construct complex patterns by combining simple patterns with logical operators. Logical AND is specified by simply concatenating two simple patterns, for instance “~C mutt-dev ~s bug”. Logical OR is specified by inserting a vertical bar (“|”) between two patterns, for instance “~C mutt-dev | ~s bug”. Additionally, you can negate a pattern by prepending a bang (“!”) character. For logical grouping, use braces (“()”). Example: “!(~t mutt|~c mutt) ~f elkins”.

### **Simple Patterns**

Mutt understands the following simple patterns:

~A	all messages
~b <i>EXPR</i>	messages which contain <i>EXPR</i> in the message body.
=b <i>STRING</i>	If IMAP is enabled, like ~b but searches for <i>STRING</i> on the server, rather than downloading each message and searching it locally.
~B <i>EXPR</i>	messages which contain <i>EXPR</i> in the whole message.
=B <i>STRING</i>	If IMAP is enabled, like ~B but searches for <i>STRING</i> on the server, rather than downloading each message and searching it locally.
~c <i>EXPR</i>	messages carbon-copied to <i>EXPR</i>
%c <i>GROUP</i>	messages carbon-copied to any member of <i>GROUP</i>
~C <i>EXPR</i>	messages either to: or cc: <i>EXPR</i>
%C <i>GROUP</i>	messages either to: or cc: to any member of <i>GROUP</i>
~d <i>MIN-MAX</i>	messages with “date-sent” in a Date range
~D	deleted messages
~e <i>EXPR</i>	messages which contain <i>EXPR</i> in the “Sender” field
%e <i>GROUP</i>	messages which contain a member of <i>GROUP</i> in the “Sender” field
~E	expired messages
~f <i>EXPR</i>	messages originating from <i>EXPR</i>
%f <i>GROUP</i>	messages originating from any member of <i>GROUP</i>
~F	flagged messages
~g	PGP signed messages
~G	PGP encrypted messages
~h <i>EXPR</i>	messages which contain <i>EXPR</i> in the message header
=h <i>STRING</i>	If IMAP is enabled, like ~h but searches for <i>STRING</i> on the server, rather than downloading each message and searching it locally. <i>STRING</i> must be of the form “header: substring”

<code>~H <i>EXPR</i></code>	messages with spam tags matching <i>EXPR</i>
<code>~i <i>EXPR</i></code>	messages which match <i>EXPR</i> in the “Message-ID” field
<code>~k</code>	messages containing PGP key material
<code>~l</code>	messages addressed to a known mailing list (defined by either <b>subscribe</b> or <b>list</b> )
<code>~L <i>EXPR</i></code>	messages either originated or received by <i>EXPR</i>
<code>%L <i>GROUP</i></code>	messages either originated or received by any member of <i>GROUP</i>
<code>~m <i>MIN-MAX</i></code>	message in the range <i>MIN</i> to <i>MAX</i>
<code>~M <i>EXPR</i></code>	messages which contain a mime Content-Type matching <i>EXPR</i>
<code>~n <i>MIN-MAX</i></code>	messages with a score in the range <i>MIN</i> to <i>MAX</i>
<code>~N</code>	new messages
<code>~O</code>	old messages
<code>~p</code>	messages addressed to you (consults \$from, <b>alternates</b> , and local account/hostname information)
<code>~P</code>	messages from you (consults \$from, <b>alternates</b> , and local account/hostname information)
<code>~Q</code>	messages which have been replied to
<code>~r <i>MIN-MAX</i></code>	messages with “date-received” in a Date range
<code>~R</code>	read messages
<code>~s <i>EXPR</i></code>	messages having <i>EXPR</i> in the “Subject” field.
<code>~S</code>	superseded messages
<code>~t <i>EXPR</i></code>	messages addressed to <i>EXPR</i>
<code>~T</code>	tagged messages
<code>~u</code>	messages addressed to a subscribed mailing list (defined by <b>subscribe</b> commands)
<code>~U</code>	unread messages
<code>~v</code>	message is part of a collapsed thread.
<code>~V</code>	cryptographically verified messages
<code>~x <i>EXPR</i></code>	messages which contain <i>EXPR</i> in the “References” or “In-Reply-To” field
<code>~X <i>MIN-MAX</i></code>	messages with <i>MIN</i> - <i>MAX</i> attachments
<code>~y <i>EXPR</i></code>	messages which contain <i>EXPR</i> in the “X-Label” field
<code>~z <i>MIN-MAX</i></code>	messages with a size in the range <i>MIN</i> to <i>MAX</i>
<code>~=</code>	duplicated messages (see \$duplicate_threads)
<code>~\$</code>	unreferenced message (requires threaded view)
<code>~(PATTERN)</code>	messages in threads containing messages matching a certain pattern, e.g. all threads containing messages from you: <code>~(P)</code>
<code>~&lt;(PATTERN)</code>	messages whose immediate parent matches PATTERN, e.g. replies to your messages: <code>~&lt;(P)</code>
<code>~&gt;(PATTERN)</code>	messages having an immediate child matching PATTERN, e.g. messages you replied to: <code>~&gt;(P)</code>

In the above, *EXPR* is a regular expression.

With the `~d`, `~m`, `~n`, `~r`, `~X`, and `~z` modifiers, you can also specify ranges in the forms `<MAX`, `>MIN`, `MIN-`, and `-MAX`.

With the `~z` modifier, the suffixes “K” and “M” are allowed to specify kilobyte and megabyte respectively.

The `~b`, `~B`, `~h`, `~M`, and `~X` modifiers require reading each message in, which can be much slower.

You can force Mutt to treat *EXPR* as a simple string instead of a regular expression by using `=` instead of `~` in the pattern name.



### Matching dates

The `~d` and `~r` modifiers are used to match date ranges, which are interpreted to be given in your local time zone.

A date is of the form `DD[/MM[/[cc]YY]]`, that is, a two-digit date, optionally followed by a two-digit month, optionally followed by a year specifications. Omitted fields default to the current month and year.

Mutt understands either two or four digit year specifications. When given a two-digit year, mutt will interpret values less than 70 as lying in the 21st century (i.e., “38” means 2038 and not 1938, and “00” is interpreted as 2000), and values greater than or equal to 70 as lying in the 20th century.

Note that this behavior *is* Y2K compliant, but that mutt *does* have a Y2.07K problem.

Alternatively, you may use `YYYYMMDD` to specify a date.

If a date range consists of a single date, the modifier in question will match that precise date. If the date range consists of a dash (“-”), followed by a date, this range will match any date before and up to the date given. Similarly, a date followed by a dash matches the date given and any later point of time. Two dates, separated by a dash, match any date which lies in the given range of time.

You can also modify any absolute date by giving an error range. An error range consists of one of the characters `+`, `-`, `*`, followed by a positive number, followed by one of the unit characters `y`, `m`, `w`, or `d`, specifying a unit of years, months, weeks, or days. `+` increases the maximum date matched by the given interval of time, `-` decreases the minimum date matched by the given interval of time, and `*` increases the maximum date and decreases the minimum date matched by the given interval of time. It is possible to give multiple error margins, which cumulate. Example: `1/1/2001-1w+2w*3d`

You can also specify offsets relative to the current date. An offset is specified as one of the characters `<`, `>`, `=`, followed by a positive number, followed by one of the unit characters `y`, `m`, `w`, `d`, `H`, `M`, or `S`. `>` matches dates which are older than the specified amount of time, an offset which begins with the character `<` matches dates which are more recent than the specified amount of time, and an offset which begins with the character `=` matches points of time which are precisely the given amount of time ago.

## CONFIGURATION VARIABLES

### **abort\_noattach**

Type: quadoption  
Default: no

When the body of the message matches `$abort_noattach_regexp` and there are no attachments, this quadoption controls whether to abort sending the message.

### **abort\_noattach\_regexp**

Type: regular expression  
Default: “attach”

Specifies a regular expression to match against the body of the message, to determine if an attachment was mentioned but mistakenly forgotten. If it matches, `$abort_noattach` will be consulted to determine if message sending will be aborted.

Like other regular expressions in Mutt, the search is case sensitive if the pattern contains at least one upper case letter, and case insensitive otherwise.

**abort\_nosubject**

Type: quadoption  
Default: ask–yes

If set to *yes*, when composing messages and no subject is given at the subject prompt, composition will be aborted. If set to *no*, composing messages with no subject given at the subject prompt will never be aborted.

**abort\_unmodified**

Type: quadoption  
Default: yes

If set to *yes*, composition will automatically abort after editing the message body if no changes are made to the file (this check only happens after the *first* edit of the file). When set to *no*, composition will never be aborted.

**alias\_file**

Type: path  
Default: “~/muttrc”

The default file in which to save aliases created by the **<create-alias>** function. Entries added to this file are encoded in the character set specified by \$config\_charset if it is *set* or the current character set otherwise.

**Note:** Mutt will not automatically source this file; you must explicitly use the “source” command for it to be executed in case this option points to a dedicated alias file.

The default for this option is the currently used muttrc file, or “~/muttrc” if no user muttrc was found.

**alias\_format**

Type: string  
Default: “%4n %2f %t %-10a %r”

Specifies the format of the data displayed for the “alias” menu. The following **printf(3)**–style sequences are available:

%a	alias name
%f	flags – currently, a “d” for an alias marked for deletion
%n	index number
%r	address which alias expands to
%t	character which indicates if the alias is tagged for inclusion

**allow\_8bit**

Type: boolean  
Default: yes

Controls whether 8-bit data is converted to 7-bit using either Quoted-Printable or Base64 encoding when sending mail.

**allow\_ansi**

Type: boolean

Default: no

Controls whether ANSI color codes in messages (and color tags in rich text messages) are to be interpreted. Messages containing these codes are rare, but if this option is *set*, their text will be colored accordingly. Note that this may override your color choices, and even present a security problem, since a message could include a line like

```
[-- PGP output follows ...
```

and give it the same color as your attachment color (see also `$crypt_timestamp`).

**arrow\_cursor**

Type: boolean

Default: no

When *set*, an arrow (“->”) will be used to indicate the current entry in menus instead of highlighting the whole line. On slow network or modem links this will make response faster because there is less that has to be redrawn on the screen when moving to the next or previous entries in the menu.

**ascii\_chars**

Type: boolean

Default: no

If *set*, Mutt will use plain ASCII characters when displaying thread and attachment trees, instead of the default ACS characters.

**askbcc**

Type: boolean

Default: no

If *set*, Mutt will prompt you for blind-carbon-copy (Bcc) recipients before editing an outgoing message.

**askcc**

Type: boolean

Default: no

If *set*, Mutt will prompt you for carbon-copy (Cc) recipients before editing the body of an outgoing message.

### **assumed\_charset**

Type: string

Default: ""

This variable is a colon-separated list of character encoding schemes for messages without character encoding indication. Header field values and message body content without character encoding indication would be assumed that they are written in one of this list. By default, all the header fields and message body without any charset indication are assumed to be in "us-ascii".

For example, Japanese users might prefer this:

```
set assumed_charset="iso-2022-jp:euc-jp:shift_jis:utf-8"
```

However, only the first content is valid for the message body.

### **attach\_charset**

Type: string

Default: ""

This variable is a colon-separated list of character encoding schemes for text file attachments. Mutt uses this setting to guess which encoding files being attached are encoded in to convert them to a proper character set given in \$send\_charset.

If *unset*, the value of \$charset will be used instead. For example, the following configuration would work for Japanese text handling:

```
set attach_charset="iso-2022-jp:euc-jp:shift_jis:utf-8"
```

Note: for Japanese users, "iso-2022-\*" must be put at the head of the value as shown above if included.

### **attach\_format**

Type: string

Default: "%u%D%I %t%4n %T%.40d%> [%.7m/%.10M, %.6e%?C?, %C?, %s]"

This variable describes the format of the "attachment" menu. The following **printf(3)**-style sequences are understood:

%C	charset
%c	requires charset conversion ("n" or "c")
%D	deleted flag

%d	description (if none, falls back to %F)
%e	MIME content-transfer-encoding
%F	filename in content-disposition header (if none, falls back to %f)
%f	filename
%I	disposition (“T” for inline, “A” for attachment)
%m	major MIME type
%M	MIME subtype
%n	attachment number
%Q	“Q”, if MIME part qualifies for attachment counting
%s	size (see formatstrings-size)
%t	tagged flag
%T	graphic tree characters
%u	unlink (=to delete) flag
%X	number of qualifying MIME parts in this part and its children (please see the “attachments” section for possible speed effects)
%>X	right justify the rest of the string and pad with character “X”
% X	pad to the end of the line with character “X”
%*X	soft-fill with character “X” as pad

For an explanation of “soft-fill”, see the `$index_format` documentation.

#### **attach\_save\_dir**

Type: path  
Default: “”

The default directory to save attachments from the “attachment” menu. If it doesn’t exist, Mutt will prompt to create the directory before saving.

If the path is invalid (e.g. not a directory, or cannot be `chdir`’ed to), Mutt will fall back to using the current directory.

#### **attach\_sep**

Type: string  
Default: “\n”

The separator to add between attachments when operating (saving, printing, piping, etc) on a list of tagged attachments.

#### **attach\_split**

Type: boolean  
Default: yes

If this variable is *unset*, when operating (saving, printing, piping, etc) on a list of tagged attachments, Mutt will concatenate the attachments and will operate on them as a single attachment. The `$attach_sep` separator is added after each attachment. When *set*, Mutt will operate on the attachments one by one.

**attribution**

Type: string (localized)  
Default: “On %d, %n wrote:”

This is the string that will precede a message which has been included in a reply. For a full listing of defined **printf(3)**–like sequences see the section on `$index_format`.

**attribution\_locale**

Type: string  
Default: “”

The locale used by **strftime(3)** to format dates in the attribution string. Legal values are the strings your system accepts for the locale environment variable `$LC_TIME`.

This variable is to allow the attribution date format to be customized by recipient or folder using hooks. By default, Mutt will use your locale environment, so there is no need to set this except to override that default.

**auto\_subscribe**

Type: boolean  
Default: no

When *set*, Mutt assumes the presence of a List–Post header means the recipient is subscribed to the list. Unless the mailing list is in the “unsubscribe” or “unlist” lists, it will be added to the “subscribe” list. Parsing and checking these things slows header reading down, so this option is disabled by default.

**auto\_tag**

Type: boolean  
Default: no

When *set*, functions in the *index* menu which affect a message will be applied to all tagged messages (if there are any). When unset, you must first use the `<tag-prefix>` function (bound to “;” by default) to make the next function apply to all tagged messages.

**autocrypt**

Type: boolean  
Default: no

When *set*, enables autocrypt, which provides passive encryption protection with keys exchanged via headers. See “autocryptdoc” for more details. (Autocrypt only)

**autocrypt\_acct\_format**

Type: string

Default: “%4n %-30a %20p %10s”

This variable describes the format of the “autocrypt account” menu. The following **printf(3)**–style sequences are understood

%a	email address
%k	gpg keyid
%n	current entry number
%p	prefer-encrypt flag
%s	status flag (active/inactive)

(Autocrypt only)

**autocrypt\_dir**

Type: path

Default: “~/mutt/autocrypt”

This variable sets where autocrypt files are stored, including the GPG keyring and sqlite database. See “autocryptdoc” for more details. (Autocrypt only)

**autocrypt\_reply**

Type: boolean

Default: yes

When *set*, replying to an autocrypt email automatically enables autocrypt in the reply. You may want to unset this if you’re using the same key for autocrypt as normal web-of-trust, so that autocrypt isn’t forced on for all encrypted replies. (Autocrypt only)

**autoedit**

Type: boolean

Default: no

When *set* along with `$edit_headers`, Mutt will skip the initial send-menu (prompting for subject and recipients) and allow you to immediately begin editing the body of your message. The send-menu may still be accessed once you have finished editing the body of your message.

**Note:** when this option is *set*, you cannot use send-hooks that depend on the recipients when composing a new (non-reply) message, as the initial list of recipients is empty.

Also see `$fast_reply`.

**background\_edit**

Type: boolean

Default: no

When *set*, Mutt will run \$editor in the background during message composition. A landing page will display, waiting for the \$editor to exit. The landing page may be exited, allowing perusal of the mailbox, or even for other messages to be composed. Backgrounded sessions may be returned to via the **<background-compose-menu>** function.

For background editing to work properly, \$editor must be set to an editor that does not try to use the Mutt terminal: for example a graphical editor, or a script launching (and waiting for) the editor in another Gnu Screen window.

For more details, see “bgedit” (“Background Editing” in the manual).

### **background\_confirm\_quit**

Type: boolean

Default: yes

When *set*, if there are any background edit sessions, you will be prompted to confirm exiting Mutt, in addition to the \$quit prompt.

### **background\_format**

Type: string

Default: “%10S %7p %s”

This variable describes the format of the “background compose” menu. The following **printf(3)**-style sequences are understood:

%i	parent message id (for replies and forwarded messages)
%n	the running number on the menu
%p	pid of the \$editor process
%r	comma separated list of “To:” recipients
%R	comma separated list of “Cc:” recipients
%s	subject of the message
%S	status of the \$editor process: running/finished

### **beep**

Type: boolean

Default: yes

When this variable is *set*, mutt will beep when an error occurs.

### **beep\_new**

Type: boolean

Default: no

When this variable is *set*, mutt will beep whenever it prints a message notifying you of new mail. This is independent of the setting of the \$beep variable.



**bounce**

Type: quadoption  
Default: ask–yes

Controls whether you will be asked to confirm bouncing messages. If set to *yes* you don't get asked if you want to bounce a message. Setting this variable to *no* is not generally useful, and thus not recommended, because you are unable to bounce messages.

**bounce\_delivered**

Type: boolean  
Default: yes

When this variable is *set*, mutt will include Delivered–To headers when bouncing messages. Postfix users may wish to *unset* this variable.

**braille\_friendly**

Type: boolean  
Default: no

When this variable is *set*, mutt will place the cursor at the beginning of the current line in menus, even when the `$arrow_cursor` variable is *unset*, making it easier for blind persons using Braille displays to follow these menus. The option is *unset* by default because many visual terminals don't permit making the cursor invisible.

**browser\_abbreviate\_mailboxes**

Type: boolean  
Default: yes

When this variable is *set*, mutt will abbreviate mailbox names in the browser mailbox list, using `'~'` and `'='` shortcuts.

The default **"alpha"** setting of `$sort_browser` uses locale–based sorting (using `strcoll(3)`), which ignores some punctuation. This can lead to some situations where the order doesn't make intuitive sense. In those cases, it may be desirable to *unset* this variable.

**browser\_sticky\_cursor**

Type: boolean  
Default: yes

When this variable is *set*, the browser will attempt to keep the cursor on the same mailbox when performing various functions. These include moving up a directory, toggling between mailboxes and directory listing, creating/renaming a mailbox, toggling subscribed mailboxes, and entering a new mask.

**certificate\_file**

Type: path

Default: “~/mutt\_certificates”

This variable specifies the file where the certificates you trust are saved. When an unknown certificate is encountered, you are asked if you accept it or not. If you accept it, the certificate can also be saved in this file and further connections are automatically accepted.

You can also manually add CA certificates in this file. Any server certificate that is signed with one of these CA certificates is also automatically accepted.

Example:

```
set certificate_file=~/.mutt/certificates
```

**change\_folder\_next**

Type: boolean

Default: no

When this variable is *set*, the **<change-folder>** function mailbox suggestion will start at the next folder in your “mailboxes” list, instead of starting at the first folder in the list.

**charset**

Type: string

Default: “”

Character set your terminal uses to display and enter textual data. It is also the fallback for \$send\_charset.

Upon startup Mutt tries to derive this value from environment variables such as **\$LC\_CTYPE** or **\$LANG**.

**Note:** It should only be set in case Mutt isn’t able to determine the character set used correctly.

**check\_mbox\_size**

Type: boolean

Default: no

When this variable is *set*, mutt will use file size attribute instead of access time when checking for new mail in mbox and mmdbf folders.

This variable is *unset* by default and should only be enabled when new mail detection for these folder types is unreliable or doesn’t work.

Note that enabling this variable should happen before any “mailboxes” directives occur in configuration files regarding mbox or mmdbf folders because mutt needs to determine the initial new mail

status of such a mailbox by performing a fast mailbox scan when it is defined. Afterwards the new mail status is tracked by file size changes.

**check\_new**

Type: boolean

Default: yes

**Note:** this option only affects *maildir* and *MH* style mailboxes.

When *set*, Mutt will check for new mail delivered while the mailbox is open. Especially with MH mailboxes, this operation can take quite some time since it involves scanning the directory and checking each file to see if it has already been looked at. If this variable is *unset*, no check for new mail is performed while the mailbox is open.

**collapse\_unread**

Type: boolean

Default: yes

When *unset*, Mutt will not collapse a thread if it contains any unread messages.

**compose\_format**

Type: string (localized)

Default: “— Mutt: Compose [Approx. msg size: %l Atts: %a]%>—”

Controls the format of the status line displayed in the “compose” menu. This string is similar to `$status_format`, but has its own set of **printf(3)**–like sequences:

%a	total number of attachments
%h	local hostname
%l	approximate size (in bytes) of the current message (see <code>formatstrings-size</code> )
%v	Mutt version string

See the text describing the `$status_format` option for more information on how to set `$compose_format`.

**config\_charset**

Type: string

Default: “”

When defined, Mutt will recode commands in rc files from this encoding to the current character set as specified by `$charset` and aliases written to `$alias_file` from the current character set.

Please note that if setting `$charset` it must be done before setting `$config_charset`.

Recoding should be avoided as it may render unconvertable characters as question marks which can lead to undesired side effects (for example in regular expressions).

**confirmappend**

Type: boolean  
Default: yes

When *set*, Mutt will prompt for confirmation when appending messages to an existing mailbox.

**confirmcreate**

Type: boolean  
Default: yes

When *set*, Mutt will prompt for confirmation when saving messages to a mailbox which does not yet exist before creating it.

**connect\_timeout**

Type: number  
Default: 30

Causes Mutt to timeout a network connection (for IMAP, POP or SMTP) after this many seconds if the connection is not able to be established. A negative value causes Mutt to wait indefinitely for the connection attempt to succeed.

**content\_type**

Type: string  
Default: “text/plain”

Sets the default Content-Type for the body of newly composed messages.

**copy**

Type: quadoption  
Default: yes

This variable controls whether or not copies of your outgoing messages will be saved for later references. Also see \$record, \$save\_name, \$force\_name and “fcc-hook”.

**copy\_decode\_weed**

Type: boolean  
Default: no

Controls whether Mutt will weed headers when invoking the **<decode-copy>** or **<decode-save>** functions.

**count\_alternatives**

Type: boolean

Default: no

When *set*, Mutt will recurse inside multipart/alternatives while performing attachment searching and counting (see attachments).

Traditionally, multipart/alternative parts have simply represented different encodings of the main content of the email. Unfortunately, some mail clients have started to place email attachments inside one of alternatives. Setting this will allow Mutt to find and count matching attachments hidden there, and include them in the index via %X or through ~X pattern matching.

**cursor\_overlay**

Type: boolean

Default: no

When *set*, Mutt will overlay the indicator, tree, sidebar\_highlight, and sidebar\_indicator colors onto the currently selected line. This will allow **default** colors in those to be overridden, and for attributes to be merged between the layers.

**crypt\_autoencrypt**

Type: boolean

Default: no

Setting this variable will cause Mutt to always attempt to PGP encrypt outgoing messages. This is probably only useful in connection to the “send-hook” command. It can be overridden by use of the pgp menu, when encryption is not required or signing is requested as well. If \$smime\_is\_default is *set*, then OpenSSL is used instead to create S/MIME messages and settings can be overridden by use of the smime menu instead. (Crypto only)

**crypt\_autopgp**

Type: boolean

Default: yes

This variable controls whether or not mutt may automatically enable PGP encryption/signing for messages. See also \$crypt\_autoencrypt, \$crypt\_replyencrypt, \$crypt\_autosign, \$crypt\_replysign and \$smime\_is\_default.

**crypt\_autosign**

Type: boolean

Default: no

Setting this variable will cause Mutt to always attempt to cryptographically sign outgoing messages. This can be overridden by use of the pgp menu, when signing is not required or encryption is requested as well. If \$smime\_is\_default is *set*, then OpenSSL is used instead to create S/MIME messages and settings can be overridden by use of the smime menu instead of the pgp menu.

(Crypto only)

**crypt\_autosmime**

Type: boolean

Default: yes

This variable controls whether or not mutt may automatically enable S/MIME encryption/signing for messages. See also `$crypt_autoencrypt`, `$crypt_replyencrypt`, `$crypt_autosign`, `$crypt_replysign` and `$smime_is_default`.

**crypt\_confirmhook**

Type: boolean

Default: yes

If set, then you will be prompted for confirmation of keys when using the `crypt-hook` command. If unset, no such confirmation prompt will be presented. This is generally considered unsafe, especially where typos are concerned.

**crypt\_opportunistic\_encrypt**

Type: boolean

Default: no

Setting this variable will cause Mutt to automatically enable and disable encryption, based on whether all message recipient keys can be located by Mutt.

When this option is enabled, Mutt will enable/disable encryption each time the TO, CC, and BCC lists are edited. If `$edit_headers` is set, Mutt will also do so each time the message is edited.

While this is set, encryption can't be manually enabled/disabled. The pgp or smime menus provide a selection to temporarily disable this option for the current message.

If `$crypt_autoencrypt` or `$crypt_replyencrypt` enable encryption for a message, this option will be disabled for that message. It can be manually re-enabled in the pgp or smime menus. (Crypto only)

**crypt\_opportunistic\_encrypt\_strong\_keys**

Type: boolean

Default: no

When set, this modifies the behavior of `$crypt_opportunistic_encrypt` to only search for "strong keys", that is, keys with full validity according to the web-of-trust algorithm. A key with marginal or no validity will not enable opportunistic encryption.

For S/MIME, the behavior depends on the backend. Classic S/MIME will filter for certificates with the 't' (trusted) flag in the .index file. The GPGME backend will use the same filters as with OpenPGP, and depends on GPGME's logic for assigning the `GPGME_VALIDITY_FULL` and

GPGME\_VALIDITY\_ULTIMATE validity flag.

#### **crypt\_protected\_headers\_read**

Type: boolean

Default: yes

When set, Mutt will display protected headers in the pager, and will update the index and header cache with revised headers. Protected headers are stored inside the encrypted or signed part of an email, to prevent disclosure or tampering. For more information see <https://github.com/autocrypt/protected-headers>. Currently Mutt only supports the Subject header.

Encrypted messages using protected headers often substitute the exposed Subject header with a dummy value (see \$crypt\_protected\_headers\_subject). Mutt will update its concept of the correct subject **after** the message is opened, i.e. via the <display-message> function. If you reply to a message before opening it, Mutt will end up using the dummy Subject header, so be sure to open such a message first. (Crypto only)

#### **crypt\_protected\_headers\_save**

Type: boolean

Default: no

When \$crypt\_protected\_headers\_read is set, and a message with a protected Subject is opened, Mutt will save the updated Subject into the header cache by default. This allows searching/limiting based on the protected Subject header if the mailbox is re-opened, without having to re-open the message each time. However, for mbox/mh mailbox types, or if header caching is not set up, you would need to re-open the message each time the mailbox was reopened before you could see or search/limit on the protected subject again.

When this variable is set, Mutt additionally saves the protected Subject back **in the clear-text message headers**. This provides better usability, but with the tradeoff of reduced security. The protected Subject header, which may have previously been encrypted, is now stored in clear-text in the message headers. Copying the message elsewhere, via Mutt or external tools, could expose this previously encrypted data. Please make sure you understand the consequences of this before you enable this variable. (Crypto only)

#### **crypt\_protected\_headers\_subject**

Type: string

Default: “...”

When \$crypt\_protected\_headers\_write is set, and the message is marked for encryption, this will be substituted into the Subject field in the message headers. To prevent a subject from being substituted, unset this variable, or set it to the empty string. (Crypto only)

#### **crypt\_protected\_headers\_write**

Type: boolean

Default: no

When set, Mutt will generate protected headers for signed and encrypted emails. Protected headers are stored inside the encrypted or signed part of an email, to prevent disclosure or tampering. For more information see <https://github.com/autocrypt/protected-headers>. Currently Mutt only supports the Subject header. (Crypto only)

**crypt\_replyencrypt**

Type: boolean

Default: yes

If *set*, automatically PGP or OpenSSL encrypt replies to messages which are encrypted. (Crypto only)

**crypt\_replysign**

Type: boolean

Default: no

If *set*, automatically PGP or OpenSSL sign replies to messages which are signed.

**Note:** this does not work on messages that are encrypted *and* signed! (Crypto only)

**crypt\_replysignencrypted**

Type: boolean

Default: no

If *set*, automatically PGP or OpenSSL sign replies to messages which are encrypted. This makes sense in combination with `$crypt_replyencrypt`, because it allows you to sign all messages which are automatically encrypted. This works around the problem noted in `$crypt_replysign`, that mutt is not able to find out whether an encrypted message is also signed. (Crypto only)

**crypt\_timestamp**

Type: boolean

Default: yes

If *set*, mutt will include a time stamp in the lines surrounding PGP or S/MIME output, so spoofing such lines is more difficult. If you are using colors to mark these lines, and rely on these, you may *unset* this setting. (Crypto only)

**crypt\_use\_gpgme**

Type: boolean

Default: no

This variable controls the use of the GPGME-enabled crypto backends. If it is *set* and Mutt was built with gpgme support, the gpgme code for S/MIME and PGP will be used instead of the classic code. Note that you need to set this option in `.muttrc`; it won't have any effect when used



interactively.

Note that the GPGME backend does not support creating old-style inline (traditional) PGP encrypted or signed messages (see `$pgp_autoinline`).

#### **crypt\_use\_pka**

Type: boolean

Default: no

Controls whether mutt uses PKA (see <http://www.g10code.de/docs/pka-intro.de.pdf>) during signature verification (only supported by the GPGME backend).

#### **crypt\_verify\_sig**

Type: quadoption

Default: yes

If “yes”, always attempt to verify PGP or S/MIME signatures. If “ask-\*”, ask whether or not to verify the signature. If “no”, never attempt to verify cryptographic signatures. (Crypto only)

#### **date\_format**

Type: string

Default: “!%a, %b %d, %Y at %I:%M:%S%p %Z”

This variable controls the format of the date printed by the “%d” sequence in `$index_format`. This is passed to the **strftime(3)** function to process the date, see the man page for the proper syntax.

Unless the first character in the string is a bang (“!”), the month and week day names are expanded according to the locale. If the first character in the string is a bang, the bang is discarded, and the month and week day names in the rest of the string are expanded in the *C* locale (that is in US English).

#### **default\_hook**

Type: string

Default: “~f %s !~P | (~P ~C %s)”

This variable controls how “message-hook”, “reply-hook”, “send-hook”, “send2-hook”, “save-hook”, and “fcc-hook” will be interpreted if they are specified with only a simple regexp, instead of a matching pattern. The hooks are expanded when they are declared, so a hook will be interpreted according to the value of this variable at the time the hook is declared.

The default value matches if the message is either from a user matching the regular expression given, or if it is from you (if the from address matches “alternates”) and is to or cc’ed to a user matching the given regular expression.

**delete**

Type: quadoption  
Default: ask–yes

Controls whether or not messages are really deleted when closing or synchronizing a mailbox. If set to *yes*, messages marked for deleting will automatically be purged without prompting. If set to *no*, messages marked for deletion will be kept in the mailbox.

**delete\_untag**

Type: boolean  
Default: yes

If this option is *set*, mutt will untag messages when marking them for deletion. This applies when you either explicitly delete a message, or when you save it to another folder.

**digest\_collapse**

Type: boolean  
Default: yes

If this option is *set*, mutt's received–attachments menu will not show the subparts of individual messages in a multipart/digest. To see these subparts, press “v” on that menu.

**display\_filter**

Type: path  
Default: “”

When set, specifies a command used to filter messages. When a message is viewed it is passed as standard input to `$display_filter`, and the filtered message is read from the standard output.

**dotlock\_program**

Type: path  
Default: “/usr/bin/mutt\_dotlock”

Contains the path of the **mutt\_dotlock(1)** binary to be used by mutt.

**dsn\_notify**

Type: string  
Default: “”

This variable sets the request for when notification is returned. The string consists of a comma separated list (no spaces!) of one or more of the following: *never*, to never request notification, *failure*, to request notification on transmission failure, *delay*, to be notified of message delays, *success*, to be notified of successful transmission.

Example:

```
set dsn_notify="failure,delay"
```

**Note:** when using `$sendmail` for delivery, you should not enable this unless you are either using Sendmail 8.8.x or greater or a MTA providing a **sendmail(1)**–compatible interface supporting the `–N` option for DSN. For SMTP delivery, DSN support is auto–detected so that it depends on the server whether DSN will be used or not.

### **dsn\_return**

Type: string

Default: ""

This variable controls how much of your message is returned in DSN messages. It may be set to either *hdrs* to return just the message header, or *full* to return the full message.

Example:

```
set dsn_return=hdrs
```

**Note:** when using `$sendmail` for delivery, you should not enable this unless you are either using Sendmail 8.8.x or greater or a MTA providing a **sendmail(1)**–compatible interface supporting the `–R` option for DSN. For SMTP delivery, DSN support is auto–detected so that it depends on the server whether DSN will be used or not.

### **duplicate\_threads**

Type: boolean

Default: yes

This variable controls whether mutt, when `$sort` is set to *threads*, threads messages with the same Message–Id together. If it is *isset*, it will indicate that it thinks the y are duplicates of each other with an equals sign in the thread tree.

### **edit\_headers**

Type: boolean

Default: no

This option allows you to edit the header of your outgoing messages along with the body of your message.

Although the compose menu may have localized header labels, the labels passed to your editor will be standard RFC 2822 headers, (e.g. To:, Cc:, Subject:). Headers added in your editor must also be RFC 2822 headers, or one of the pseudo headers listed in “edit-header”. Mutt will not understand localized header labels, just as it would not when parsing an actual email.

**Note** that changes made to the References: and Date: headers are ignored for interoperability reasons.

### **editor**

Type: path

Default: ""

This variable specifies which editor is used by mutt. It defaults to the value of the **\$VISUAL**, or **\$EDITOR**, environment variable, or to the string "vi" if neither of those are set.

The **\$editor** string may contain a *%s* escape, which will be replaced by the name of the file to be edited. If the *%s* escape does not appear in **\$editor**, a space and the name to be edited are appended.

The resulting string is then executed by running

```
sh -c 'string'
```

where *string* is the expansion of **\$editor** described above.

### **encode\_from**

Type: boolean

Default: no

When *set*, mutt will quoted-printable encode messages when they contain the string "From " (note the trailing space) in the beginning of a line. This is useful to avoid the tampering certain mail delivery and transport agents tend to do with messages (in order to prevent tools from misinterpreting the line as a mbox message separator).

### **entropy\_file**

Type: path

Default: ""

The file which includes random data that is used to initialize SSL library functions.

### **envelope\_from\_address**

Type: e-mail address

Default: ""

Manually sets the *envelope* sender for outgoing messages. This value is ignored if **\$use\_envelope\_from** is *unset*.

**error\_history**

Type: number  
Default: 30

This variable controls the size (in number of strings remembered) of the error messages displayed by mutt. These can be shown with the **<error-history>** function. The history is cleared each time this variable is set.

**escape**

Type: string  
Default: “~”

Escape character to use for functions in the built-in editor.

**fast\_reply**

Type: boolean  
Default: no

When *set*, the initial prompt for recipients and subject are skipped when replying to messages, and the initial prompt for subject is skipped when forwarding messages.

**Note:** this variable has no effect when the \$autoedit variable is *set*.

**fcc\_attach**

Type: quadoption  
Default: yes

This variable controls whether or not attachments on outgoing messages are saved along with the main body of your message.

Note: \$fcc\_before\_send forces the default (set) behavior of this option.

**fcc\_before\_send**

Type: boolean  
Default: no

When this variable is *set*, FCCs will occur before sending the message. Before sending, the message cannot be manipulated, so it will be stored the exact same as sent: \$fcc\_attach and \$fcc\_clear will be ignored (using their default values).

When *unset*, the default, FCCs will occur after sending. Variables \$fcc\_attach and \$fcc\_clear will be respected, allowing it to be stored without attachments or encryption/signing if desired.

**fcc\_clear**

Type: boolean  
 Default: no

When this variable is *set*, FCCs will be stored unencrypted and unsigned, even when the actual message is encrypted and/or signed.

Note: `$fcc_before_send` forces the default (unset) behavior of this option. (PGP only)

See also `$pgp_self_encrypt`, `$mime_self_encrypt`.

**fcc\_delimiter**

Type: string  
 Default: ""

When specified, this allows the ability to Fcc to more than one mailbox. The fcc value will be split by this delimiter and Mutt will evaluate each part as a mailbox separately.

See `$record`, "fcc-hook", and "fcc-save-hook".

**flag\_safe**

Type: boolean  
 Default: no

If set, flagged messages cannot be deleted.

**folder**

Type: path  
 Default: "~/Mail"

Specifies the default location of your mailboxes. A "+" or "=" at the beginning of a pathname will be expanded to the value of this variable. Note that if you change this variable (from the default) value you need to make sure that the assignment occurs *before* you use "+" or "=" for any other variables since expansion takes place when handling the "mailboxes" command.

**folder\_format**

Type: string  
 Default: "%2C %t %N %F %l %-8.8u %-8.8g %8s %d %f"

This variable allows you to customize the file browser display to your personal taste. This string is similar to `$index_format`, but has its own set of **printf(3)**-like sequences:

%C	current file number
%d	date/time folder was last modified
%D	date/time folder was last modified using <code>\$date_format</code> .
%f	filename ("/" is appended to directory names, "@" to symbolic links and "*" to executable files)

%F	file permissions
%g	group name (or numeric gid, if missing)
%l	number of hard links
%m	number of messages in the mailbox *
%n	number of unread messages in the mailbox *
%N	N if mailbox has new mail, blank otherwise
%s	size in bytes (see formatstrings-size)
%t	“*” if the file is tagged, blank otherwise
%u	owner name (or numeric uid, if missing)
%>X	right justify the rest of the string and pad with character “X”
% X	pad to the end of the line with character “X”
/*X	soft-fill with character “X” as pad

For an explanation of “soft-fill”, see the `$index_format` documentation.

\* = can be optionally printed if nonzero

%m, %n, and %N only work for monitored mailboxes. %m requires `$mail_check_stats` to be set. %n requires `$mail_check_stats` to be set (except for IMAP mailboxes).

### **followup\_to**

Type: boolean  
Default: yes

Controls whether or not the “Mail-Followup-To:” header field is generated when sending mail. When *set*, Mutt will generate this field when you are replying to a known mailing list, specified with the “subscribe” or “lists” commands.

This field has two purposes. First, preventing you from receiving duplicate copies of replies to messages which you send to mailing lists, and second, ensuring that you do get a reply separately for any messages sent to known lists to which you are not subscribed.

The header will contain only the list’s address for subscribed lists, and both the list address and your own email address for unsubscribed lists. Without this header, a group reply to your message sent to a subscribed list will be sent to both the list and your address, resulting in two copies of the same email for you.

### **force\_name**

Type: boolean  
Default: no

This variable is similar to `$save_name`, except that Mutt will store a copy of your outgoing message by the username of the address you are sending to even if that mailbox does not exist.

Also see the `$record` variable.

**forward\_attachments**

Type: quadoption  
Default: ask-yes

When forwarding inline (i.e. `$mime_forward unset` or answered with “no” and `$forward_decode set`), attachments which cannot be decoded in a reasonable manner will be attached to the newly composed message if this quadoption is *set* or answered with “yes”.

**forward\_attribution\_intro**

Type: string (localized)  
Default: “----- Forwarded message from %f -----”

This is the string that will precede a message which has been forwarded in the main body of a message (when `$mime_forward` is *unset*). For a full listing of defined **printf(3)**-like sequences see the section on `$index_format`. See also `$attribution_locale`.

**forward\_attribution\_trailer**

Type: string (localized)  
Default: “----- End forwarded message -----”

This is the string that will follow a message which has been forwarded in the main body of a message (when `$mime_forward` is *unset*). For a full listing of defined **printf(3)**-like sequences see the section on `$index_format`. See also `$attribution_locale`.

**forward\_decode**

Type: boolean  
Default: yes

Controls the decoding of complex MIME messages into **text/plain** when forwarding a message. The message header is also RFC2047 decoded. This variable is only used, if `$mime_forward` is *unset*, otherwise `$mime_forward_decode` is used instead.

**forward\_decrypt**

Type: quadoption  
Default: yes

This quadoption controls the handling of encrypted messages when forwarding or attaching a message. When set to or answered “yes”, the outer layer of encryption is stripped off.

This variable is used if `$mime_forward` is *set* and `$mime_forward_decode` is *unset*. It is also used when attaching a message via **<attach-message>** in the compose menu. (PGP only)



**forward\_edit**

Type: quadoption  
Default: yes

This quadoption controls whether or not the user is automatically placed in the editor when forwarding messages. For those who always want to forward with no modification, use a setting of “no”.

**forward\_format**

Type: string  
Default: “[%a: %s]”

This variable controls the default subject when forwarding a message. It uses the same format sequences as the \$index\_format variable.

**forward\_quote**

Type: boolean  
Default: no

When *set*, forwarded messages included in the main body of the message (when \$mime\_forward is *unset*) will be quoted using \$indent\_string.

**from**

Type: e-mail address  
Default: “”

When *set*, this variable contains a default from address. It can be overridden using “my\_hdr” (including from a “send-hook”) and \$reverse\_name. This variable is ignored if \$use\_from is *unset*.

This setting defaults to the contents of the environment variable **\$EMAIL**.

**gecos\_mask**

Type: regular expression  
Default: “^[^,]\*”

A regular expression used by mutt to parse the GECOS field of a password entry when expanding the alias. The default value will return the string up to the first “,” encountered. If the GECOS field contains a string like “lastname, firstname” then you should set it to “.\*”.

This can be useful if you see the following behavior: you address an e-mail to user ID “stevef” whose full name is “Steve Franklin”. If mutt expands “stevef” to “Franklin” stevef@foo.bar then you should set the \$gecos\_mask to a regular expression that will match the whole name so mutt will expand “Franklin” to “Franklin, Steve”.

**hdrs**

Type: boolean  
Default: yes

When *unset*, the header fields normally added by the “my\_hdr” command are not created. This variable *must* be unset before composing a new message or replying in order to take effect. If *set*, the user defined header fields are added to every new message.

**header**

Type: boolean  
Default: no

When *set*, this variable causes Mutt to include the header of the message you are replying to into the edit buffer. The \$weed setting applies.

**header\_cache**

Type: path  
Default: “”

This variable points to the header cache database. If pointing to a directory Mutt will contain a header cache database file per folder, if pointing to a file that file will be a single global header cache. By default it is *unset* so no header caching will be used.

Header caching can greatly improve speed when opening POP, IMAP MH or Maildir folders, see “caching” for details.

**header\_cache\_compress**

Type: boolean  
Default: yes

When mutt is compiled with qdbm, tokyocabinet, or kyotocabinet as header cache backend, this option determines whether the database will be compressed. Compression results in database files roughly being one fifth of the usual disk space, but the decompression can result in a slower opening of cached folder(s) which in general is still much faster than opening non header cached folders.

**header\_cache\_pagesize**

Type: number (long)  
Default: 16384

When mutt is compiled with either gdbm or bdb4 as the header cache backend, this option changes the database page size. Too large or too small values can waste space, memory, or CPU time. The default should be more or less optimal for most use cases.

**header\_color\_partial**

Type: boolean

Default: no

When *set*, color header regexps behave like color body regexps: color is applied to the exact text matched by the regexp. When *unset*, color is applied to the entire header .

One use of this option might be to apply color to just the header labels.

See “color” for more details.

**help**

Type: boolean

Default: yes

When *set*, help lines describing the bindings for the major functions provided by each menu are displayed on the first line of the screen.

**Note:** The binding will not be displayed correctly if the function is bound to a sequence rather than a single keystroke. Also, the help line may not be updated if a binding is changed while Mutt is running. Since this variable is primarily aimed at new users, neither of these should present a major problem.

**hidden\_host**

Type: boolean

Default: no

When *set*, mutt will skip the host name part of \$hostname variable when adding the domain part to addresses. This variable does not affect the generation of Message-IDs, and it will not lead to the cut-off of first-level domains.

**hide\_limited**

Type: boolean

Default: no

When *set*, mutt will not show the presence of messages that are hidden by limiting, in the thread tree.

**hide\_missing**

Type: boolean

Default: yes

When *set*, mutt will not show the presence of missing messages in the thread tree.

**hide\_thread\_subject**

Type: boolean  
Default: yes

When *set*, mutt will not show the subject of messages in the thread tree that have the same subject as their parent or closest previously displayed sibling.

**hide\_top\_limited**

Type: boolean  
Default: no

When *set*, mutt will not show the presence of messages that are hidden by limiting, at the top of threads in the thread tree. Note that when `$hide_limited` is *set*, this option will have no effect.

**hide\_top\_missing**

Type: boolean  
Default: yes

When *set*, mutt will not show the presence of missing messages at the top of threads in the thread tree. Note that when `$hide_missing` is *set*, this option will have no effect.

**history**

Type: number  
Default: 10

This variable controls the size (in number of strings remembered) of the string history buffer per category. The buffer is cleared each time the variable is set.

**history\_file**

Type: path  
Default: “~/mutthistory”

The file in which Mutt will save its history.

Also see `$save_history`.

**history\_remove\_dups**

Type: boolean  
Default: no

When *set*, all of the string history will be scanned for duplicates when a new entry is added. Duplicate entries in the `$history_file` will also be removed when it is periodically compacted.

**honor\_disposition**

Type: boolean  
Default: no

When *set*, Mutt will not display attachments with a disposition of “attachment” inline even if it could render the part to plain text. These MIME parts can only be viewed from the attachment menu.

If *unset*, Mutt will render all MIME parts it can properly transform to plain text.

**honor\_followup\_to**

Type: quadoption  
Default: yes

This variable controls whether or not a Mail-Followup-To header is honored when group-replying to a message.

**hostname**

Type: string  
Default: “”

Specifies the fully-qualified hostname of the system mutt is running on containing the host’s name and the DNS domain it belongs to. It is used as the domain part (after “@”) for local email addresses as well as Message-Id headers.

Its value is determined at startup as follows: the node’s hostname is first determined by the **uname(3)** function. The domain is then looked up using the **gethostname(2)** and **getaddrinfo(3)** functions. If those calls are unable to determine the domain, the full value returned by **uname** is used. Optionally, Mutt can be compiled with a fixed domain name in which case a detected one is not used.

Starting in Mutt 2.0, the operations described in the previous paragraph are performed after the muttrc is processed, instead of beforehand. This way, if the DNS operations are creating delays at startup, you can avoid those by manually setting the value in your muttrc.

Also see \$use\_domain and \$hidden\_host.

**idn\_decode**

Type: boolean  
Default: yes

When *set*, Mutt will show you international domain names decoded. Note: You can use IDNs for addresses even if this is *unset*. This variable only affects decoding. (IDN only)

**idn\_encode**

Type: boolean  
Default: yes

When *set*, Mutt will encode international domain names using IDN. Unset this if your SMTP server can handle newer (RFC 6531) UTF-8 encoded domains. (IDN only)

**ignore\_linear\_white\_space**

Type: boolean  
Default: no

This option replaces linear-white-space between encoded-word and text to a single space to prevent the display of MIME-encoded “Subject:” field from being divided into multiple lines.

**ignore\_list\_reply\_to**

Type: boolean  
Default: no

Affects the behavior of the **<reply>** function when replying to messages from mailing lists (as defined by the “subscribe” or “lists” commands). When *set*, if the “Reply-To:” field is set to the same value as the “To:” field, Mutt assumes that the “Reply-To:” field was set by the mailing list to automate responses to the list, and will ignore this field. To direct a response to the mailing list when this option is *set*, use the **<list-reply>** function; **<group-reply>** will reply to both the sender and the list.

**imap\_authenticators**

Type: string  
Default: “”

This is a colon-delimited list of authentication methods mutt may attempt to use to log in to an IMAP server, in the order mutt should try them. Authentication methods are either “login” or the right side of an IMAP “AUTH=xxx” capability string, e.g. “digest-md5”, “gssapi” or “cram-md5”. This option is case-insensitive. If it’s *unset* (the default) mutt will try all available methods, in order from most-secure to least-secure.

Example:

```
set imap_authenticators="gssapi:cram-md5:login"
```

**Note:** Mutt will only fall back to other authentication methods if the previous methods are unavailable. If a method is available but authentication fails, mutt will not connect to the IMAP server.

**imap\_check\_subscribed**

Type: boolean

Default: no

When *set*, mutt will fetch the set of subscribed folders from your server on connection, and add them to the set of mailboxes it polls for new mail just as if you had issued individual “mailboxes” commands.

**imap\_condstore**

Type: boolean

Default: no

When *set*, mutt will use the CONDSTORE extension (RFC 7162) if advertised by the server. Mutt’s current implementation is basic, used only for initial message fetching and flag updates.

For some IMAP servers, enabling this will slightly speed up downloading initial messages. Unfortunately, Gmail is not one those, and displays worse performance when enabled. Your mileage may vary.

**imap\_deflate**

Type: boolean

Default: no

When *set*, mutt will use the COMPRESS=DEFLATE extension (RFC 4978) if advertised by the server.

In general a good compression efficiency can be achieved, which speeds up reading large mailboxes also on fairly good connections.

**imap\_delim\_chars**

Type: string

Default: “/.”

This contains the list of characters which you would like to treat as folder separators for displaying IMAP paths. In particular it helps in using the “=” shortcut for your *folder* variable.

**imap\_fetch\_chunk\_size**

Type: number (long)

Default: 0

When set to a value greater than 0, new headers will be downloaded in groups of this many headers per request. If you have a very large mailbox, this might prevent a timeout and disconnect when opening the mailbox, by sending a FETCH per set of this many headers, instead of a single FETCH for all new headers.

**imap\_headers**

Type: string  
Default: ""

Mutt requests these header fields in addition to the default headers ("Date:", "From:", "Sender:", "Subject:", "To:", "Cc:", "Message-Id:", "References:", "Content-Type:", "Content-Description:", "In-Reply-To:", "Reply-To:", "Lines:", "List-Post:", "X-Label:") from IMAP servers before displaying the index menu. You may want to add more headers for spam detection.

**Note:** This is a space separated list, items should be uppercase and not contain the colon, e.g. "X-BOGOSITY X-SPAM-STATUS" for the "X-Bogosity:" and "X-Spam-Status:" header fields.

**imap\_idle**

Type: boolean  
Default: no

When *set*, mutt will attempt to use the IMAP IDLE extension to check for new mail in the current mailbox. Some servers (dovecot was the inspiration for this option) react badly to mutt's implementation. If your connection seems to freeze up periodically, try unsetting this.

**imap\_keepalive**

Type: number  
Default: 300

This variable specifies the maximum amount of time in seconds that mutt will wait before polling open IMAP connections, to prevent the server from closing them before mutt has finished with them. The default is well within the RFC-specified minimum amount of time (30 minutes) before a server is allowed to do this, but in practice the RFC does get violated every now and then. Reduce this number if you find yourself getting disconnected from your IMAP server due to inactivity.

**imap\_list\_subscribed**

Type: boolean  
Default: no

This variable configures whether IMAP folder browsing will look for only subscribed folders or all folders. This can be toggled in the IMAP browser with the **<toggle-subscribed>** function.

**imap\_login**

Type: string  
Default: ""

Your login name on the IMAP server.



This variable defaults to the value of `$imap_user`.

**imap\_oauth\_refresh\_command**

Type: string

Default: ""

The command to run to generate an OAUTH refresh token for authorizing your connection to your IMAP server. This command will be run on every connection attempt that uses the OAUTH-BEARER authentication mechanism. See "oauth" for details.

**imap\_pass**

Type: string

Default: ""

Specifies the password for your IMAP account. If *unset*, Mutt will prompt you for your password when you invoke the `<imap-fetch-mail>` function or try to open an IMAP folder.

**Warning:** you should only use this option when you are on a fairly secure machine, because the superuser can read your muttrc even if you are the only one who can read the file.

**imap\_passive**

Type: boolean

Default: yes

When *set*, mutt will not open new IMAP connections to check for new mail. Mutt will only check for new mail over existing IMAP connections. This is useful if you don't want to be prompted for user/password pairs on mutt invocation, or if opening the connection is slow.

**imap\_peek**

Type: boolean

Default: yes

When *set*, mutt will avoid implicitly marking your mail as read whenever you fetch a message from the server. This is generally a good thing, but can make closing an IMAP folder somewhat slower. This option exists to appease speed freaks.

**imap\_pipeline\_depth**

Type: number

Default: 15

Controls the number of IMAP commands that may be queued up before they are sent to the server. A deeper pipeline reduces the amount of time mutt must wait for the server, and can make IMAP servers feel much more responsive. But not all servers correctly handle pipelined commands, so if you have problems you might want to try setting this variable to 0.

**Note:** Changes to this variable have no effect on open connections.

**imap\_poll\_timeout**

Type: number

Default: 15

This variable specifies the maximum amount of time in seconds that mutt will wait for a response when polling IMAP connections for new mail, before timing out and closing the connection. Set to 0 to disable timing out.

**imap\_qresync**

Type: boolean

Default: no

When *set*, mutt will use the QRESYNC extension (RFC 7162) if advertised by the server. Mutt's current implementation is basic, used only for initial message fetching and flag updates.

Note: this feature is currently experimental. If you experience strange behavior, such as duplicate or missing messages please file a bug report to let us know.

**imap\_servernoise**

Type: boolean

Default: yes

When *set*, mutt will display warning messages from the IMAP server as error messages. Since these messages are often harmless, or generated due to configuration problems on the server which are out of the users' hands, you may wish to suppress them at some point.

**imap\_user**

Type: string

Default: ""

The name of the user whose mail you intend to access on the IMAP server.

This variable defaults to your user name on the local machine.

**implicit\_autoview**

Type: boolean

Default: no

If set to "yes", mutt will look for a mailcap entry with the "**copiousoutput**" flag set for *very* MIME attachment it doesn't have an internal viewer defined for. If such an entry is found, mutt will use the viewer defined in that entry to convert the body part to text form.

**include**

Type: quadoption  
Default: ask-yes

Controls whether or not a copy of the message(s) you are replying to is included in your reply.

**include\_encrypted**

Type: boolean  
Default: no

Controls whether or not Mutt includes separately encrypted attachment contents when replying.

This variable was added to prevent accidental exposure of encrypted contents when replying to an attacker. If a previously encrypted message were attached by the attacker, they could trick an unwary recipient into decrypting and including the message in their reply.

**include\_onlyfirst**

Type: boolean  
Default: no

Controls whether or not Mutt includes only the first attachment of the message you are replying.

**indent\_string**

Type: string  
Default: "> "

Specifies the string to prepend to each line of text quoted in a message to which you are replying. You are strongly encouraged not to change this value, as it tends to agitate the more fanatical netizens.

The value of this option is ignored if \$text\_flowed is set, because the quoting mechanism is strictly defined for format=flowed.

This option is a format string, please see the description of \$index\_format for supported **printf(3)**-style sequences.

**index\_format**

Type: string  
Default: "%4C %Z % {%b %d} %-15.15L (%?l?%4l&%4c?) %s"

This variable allows you to customize the message index display to your personal taste.

"Format strings" are similar to the strings used in the C function **printf(3)** to format output (see the man page for more details). For an explanation of the %? construct, see the \$status\_format description. The following sequences are defined in Mutt:

%a	address of the author
%A	reply-to address (if present; otherwise: address of author)
%b	filename of the original message folder (think mailbox)
%B	the list to which the letter was sent, or else the folder name (%b).
%c	number of characters (bytes) in the message (see formatstrings-size)
%C	current message number
%d	date and time of the message in the format specified by \$date_format converted to sender's time zone
%D	date and time of the message in the format specified by \$date_format converted to the local time zone
%e	current message number in thread
%E	number of messages in current thread
%f	sender (address + real name), either From: or Return-Path:
%F	author name, or recipient name if the message is from you
%H	spam attribute(s) of this message
%i	message-id of the current message
%l	number of lines in the unprocessed message (may not work with maildir, mh, and IMAP folders)
%L	If an address in the "To:" or "Cc:" header field matches an address defined by the users "subscribe" command, this displays "To <list-name>", otherwise the same as %F.
%m	total number of message in the mailbox
%M	number of hidden messages if the thread is collapsed.
%N	message score
%n	author's real name (or address if missing)
%O	original save folder where mutt would formerly have stashed the message: list name or recipient name if not sent to a list
%P	progress indicator for the built-in pager (how much of the file has been displayed)
%r	comma separated list of "To:" recipients
%R	comma separated list of "Cc:" recipients
%s	subject of the message
%S	single character status of the message ("N"/"O"/"D"/"d"/"!"/"r"/"*)
%t	"To:" field (recipients)
%T	the appropriate character from the \$to_chars string
%u	user (login) name of the author
%v	first name of the author, or the recipient if the message is from you
%X	number of attachments (please see the "attachments" section for possible speed effects)
%y	"X-Label:" field, if present
%Y	"X-Label:" field, if present, and (1) not at part of a thread tree, (2) at the top of a thread, or (3) "X-Label:" is different from preceding message's "X-Label:".
%Z	a three character set of message status flags. the first character is new/read/replied flags ("n"/"o"/"r"/"O"/"N"). the second is deleted or encryption flags ("D"/"d"/"S"/"P"/"s"/"K"). the third is either tagged/flagged ("*/"!"), or one of the characters listed in \$to_chars.
%@name@	insert and evaluate format-string from the matching "index-format-hook" command
%{fmt}	the date and time of the message is converted to sender's time zone, and "fmt" is expanded by the library function <b>strftime(3)</b> ; a leading bang disables locales
%[fmt]	the date and time of the message is converted to the local time zone, and "fmt" is expanded by the library function <b>strftime(3)</b> ; a leading bang disables locales
%(fmt)	the local date and time when the message was received. "fmt" is expanded by the library function <b>strftime(3)</b> ; a leading bang disables locales

%<fmt> the current local time. “fmt” is expanded by the library function **strftime(3)**; a leading bang disables locales.

%>X right justify the rest of the string and pad with character “X”

%|X pad to the end of the line with character “X”

%\*X soft-fill with character “X” as pad

Note that for mbox/mmdf, “%l” applies to the unprocessed message, and for maildir/mh, the value comes from the “Lines:” header field when present (the meaning is normally the same). Thus the value depends on the encodings used in the different parts of the message and has little meaning in practice.

“Soft-fill” deserves some explanation: Normal right-justification will print everything to the left of the “%>”, displaying padding and whatever lies to the right only if there’s room. By contrast, soft-fill gives priority to the right-hand side, guaranteeing space to display it and showing padding only if there’s still room. If necessary, soft-fill will eat text leftwards to make room for rightward text.

Note that these expandos are supported in “save-hook”, “fcc-hook”, “fcc-save-hook”, and “index-format-hook”.

They are also supported in the configuration variables \$attribution, \$forward\_attribution\_intro, \$forward\_attribution\_trailer, \$forward\_format, \$indent\_string, \$message\_format, \$pager\_format, and \$post\_indent\_string.

## ispell

Type: path  
Default: “/usr/bin/ispell”

How to invoke ispell (GNU’s spell-checking software).

## keep\_flagged

Type: boolean  
Default: no

If *set*, read messages marked as flagged will not be moved from your spool mailbox to your \$mbox mailbox, or as a result of a “mbox-hook” command.

## local\_date\_header

Type: boolean  
Default: yes

If *set*, convert the date in the Date header of sent emails into local (sender’s) timezone.

**mail\_check**

Type: number  
Default: 5

This variable configures how often (in seconds) mutt should look for new mail. Also see the `$timeout` variable.

**mail\_check\_recent**

Type: boolean  
Default: yes

When *set*, Mutt will only notify you about new mail that has been received since the last time you opened the mailbox. When *unset*, Mutt will notify you if any new mail exists in the mailbox, regardless of whether you have visited it recently.

When *\$mark\_old* is set, Mutt does not consider the mailbox to contain new mail if only old messages exist.

**mail\_check\_stats**

Type: boolean  
Default: no

When *set*, mutt will periodically calculate message statistics of a mailbox while polling for new mail. It will check for unread, flagged, and total message counts. Because this operation is more performance intensive, it defaults to *unset*, and has a separate option, *\$mail\_check\_stats\_interval*, to control how often to update these counts.

Message statistics can also be explicitly calculated by invoking the `<check-stats>` function.

**mail\_check\_stats\_interval**

Type: number  
Default: 60

When *\$mail\_check\_stats* is *set*, this variable configures how often (in seconds) mutt will update message counts.

**mailcap\_path**

Type: string  
Default: ""

This variable specifies which files to consult when attempting to display MIME bodies not directly supported by Mutt. The default value is generated during startup: see the “mailcap” section of the manual.

**mailcap\_sanitize**

Type: boolean

Default: yes

If *set*, mutt will restrict possible characters in mailcap % expandos to a well-defined set of safe characters. This is the safe setting, but we are not sure it doesn't break some more advanced MIME stuff.

**DON'T CHANGE THIS SETTING UNLESS YOU ARE REALLY SURE WHAT YOU ARE DOING!**

**maildir\_header\_cache\_verify**

Type: boolean

Default: yes

Check for Maildir unaware programs other than mutt having modified maildir files when the header cache is in use. This incurs one **stat(2)** per message every time the folder is opened (which can be very slow for NFS folders).

**maildir\_trash**

Type: boolean

Default: no

If *set*, messages marked as deleted will be saved with the maildir trashed flag instead of unlinked. **Note:** this only applies to maildir-style mailboxes. Setting it will have no effect on other mailbox types.

**maildir\_check\_cur**

Type: boolean

Default: no

If *set*, mutt will poll both the new and cur directories of a maildir folder for new messages. This might be useful if other programs interacting with the folder (e.g. dovecot) are moving new messages to the cur directory. Note that setting this option may slow down polling for new messages in large folders, since mutt has to scan all cur messages.

**mark\_macro\_prefix**

Type: string

Default: ""

Prefix for macros created using mark-message. A new macro automatically generated with `<mark-message>a` will be composed from this prefix and the letter *a*.

**mark\_old**

Type: boolean  
Default: yes

Controls whether or not mutt marks *new* **unread** messages as *old* if you exit a mailbox without reading them. With this option *set*, the next time you start mutt, the messages will show up with an “O” next to them in the index menu, indicating that they are old.

**markers**

Type: boolean  
Default: yes

Controls the display of wrapped lines in the internal pager. If set, a “+” marker is displayed at the beginning of wrapped lines.

Also see the \$smart\_wrap variable.

**mask**

Type: regular expression  
Default: “!\\.\\.\\.”

A regular expression used in the file browser, optionally preceded by the *not* operator “!”. Only files whose names match this mask will be shown. The match is always case-sensitive.

**mbx**

Type: path  
Default: “~/mbx”

This specifies the folder into which read mail in your \$spoolfile folder will be appended.

Also see the \$move variable.

**mbx\_type**

Type: folder magic  
Default: mbx

The default mailbox type used when creating new folders. May be any of “mbx”, “MMDF”, “MH” and “Maildir”. This is overridden by the **-m** command-line option.

**menu\_context**

Type: number  
Default: 0



This variable controls the number of lines of context that are given when scrolling through menus. (Similar to `$pager_context`.)

**menu\_move\_off**

Type: boolean

Default: yes

When *unset*, the bottom entry of menus will never scroll up past the bottom of the screen, unless there are less entries than lines. When *set*, the bottom entry may move off the bottom.

**menu\_scroll**

Type: boolean

Default: no

When *set*, menus will be scrolled up or down one line when you attempt to move across a screen boundary. If *unset*, the screen is cleared and the next or previous page of the menu is displayed (useful for slow links to avoid many redraws).

**message\_cache\_clean**

Type: boolean

Default: no

If *set*, mutt will clean out obsolete entries from the message cache when the mailbox is synchronized. You probably only want to set it every once in a while, since it can be a little slow (especially for large folders).

**message\_cachedir**

Type: path

Default: ""

Set this to a directory and mutt will cache copies of messages from your IMAP and POP servers here. You are free to remove entries at any time.

When setting this variable to a directory, mutt needs to fetch every remote message only once and can perform regular expression searches as fast as for local folders.

Also see the `$message_cache_clean` variable.

**message\_format**

Type: string

Default: "%s"

This is the string displayed in the "attachment" menu for attachments of type **message/rfc822**. For a full listing of defined **printf(3)**-like sequences see the section on `$index_format`.

**message\_id\_format**

Type: string

Default: “&lt;%z@%f&gt;”

This variable describes the format of the Message-ID generated when sending messages. Mutt 2.0 introduced a more compact format, but this variable allows the ability to choose your own format. The value may end in “|” to invoke an external filter. See `formatstrings-filters`.

Please note that the Message-ID value follows a strict syntax, and you are responsible for ensuring correctness if you change this from the default. In particular, the value must follow the syntax in RFC 5322: “<” **id-left** “@” **id-right** “>”. No spaces are allowed, and **id-left** should follow the dot-atom-text syntax in the RFC. The **id-right** should generally be left at %f.

The old Message-ID format can be used by setting this to:  
“<%Y%02m%02d%02H%02M%02S.G%c%p@%f>”

The following **printf(3)**-style sequences are understood:

%c	step counter looping from “A” to “Z”
%d	current day of the month (GMT)
%f	\$hostname
%H	current hour using a 24-hour clock (GMT)
%m	current month number (GMT)
%M	current minute of the hour (GMT)
%p	pid of the running mutt process
%r	3 bytes of pseudorandom data encoded in Base64
%S	current second of the minute (GMT)
%x	1 byte of pseudorandom data hex encoded (example: '1b')
%Y	current year using 4 digits (GMT)
%z	4 byte timestamp + 8 bytes of pseudorandom data encoded in Base64

**meta\_key**

Type: boolean

Default: no

If *set*, forces Mutt to interpret keystrokes with the high bit (bit 8) set as if the user had pressed the Esc key and whatever key remains after having the high bit removed. For example, if the key pressed has an ASCII value of **0xf8**, then this is treated as if the user had pressed Esc then “x”. This is because the result of removing the high bit from **0xf8** is **0x78**, which is the ASCII character “x”.

**metoo**

Type: boolean

Default: no

If *unset*, Mutt will remove your address (see the “alternates” command) from the list of recipients when replying to a message.

**mh\_purge**

Type: boolean

Default: no

When *unset*, mutt will mimic mh's behavior and rename deleted messages to *,<old file name>* in mh folders instead of really deleting them. This leaves the message on disk but makes programs reading the folder ignore it. If the variable is *set*, the message files will simply be deleted.

This option is similar to \$maildir\_trash for Maildir folders.

**mh\_seq\_flagged**

Type: string

Default: "flagged"

The name of the MH sequence used for flagged messages.

**mh\_seq\_replied**

Type: string

Default: "replied"

The name of the MH sequence used to tag replied messages.

**mh\_seq\_unseen**

Type: string

Default: "unseen"

The name of the MH sequence used for unseen messages.

**mime\_forward**

Type: quadoption

Default: no

When *set*, the message you are forwarding will be attached as a separate **message/rfc822** MIME part instead of included in the main body of the message. This is useful for forwarding MIME messages so the receiver can properly view the message as it was delivered to you. If you like to switch between MIME and not MIME from mail to mail, set this variable to "ask-no" or "ask-yes".

Also see \$forward\_decode and \$mime\_forward\_decode.

**mime\_forward\_decode**

Type: boolean

Default: no

Controls the decoding of complex MIME messages into **text/plain** when forwarding a message while `$mime_forward` is *set*. Otherwise `$forward_decode` is used instead.

#### **mime\_forward\_rest**

Type: quadoption

Default: yes

When forwarding multiple attachments of a MIME message from the attachment menu, attachments which cannot be decoded in a reasonable manner will be attached to the newly composed message if this option is *set*.

#### **mime\_type\_query\_command**

Type: string

Default: ""

This specifies a command to run, to determine the mime type of a new attachment when composing a message. Unless `$mime_type_query_first` is set, this will only be run if the attachment's extension is not found in the `mime.types` file.

The string may contain a "%s", which will be substituted with the attachment filename. Mutt will add quotes around the string substituted for "%s" automatically according to shell quoting rules, so you should avoid adding your own. If no "%s" is found in the string, Mutt will append the attachment filename to the end of the string.

The command should output a single line containing the attachment's mime type.

Suggested values are "xdg-mime query filetype" or "file -bi".

#### **mime\_type\_query\_first**

Type: boolean

Default: no

When *set*, the `$mime_type_query_command` will be run before the `mime.types` lookup.

#### **mix\_entry\_format**

Type: string

Default: "%4n %c %-16s %a"

This variable describes the format of a remailer line on the mixmaster chain selection screen. The following **printf(3)**-like sequences are supported:

%n	The running number on the menu.
%c	ReMailer capabilities.
%s	The remailer's short name.
%a	The remailer's e-mail address.

**mixmaster**

Type: path

Default: “mixmaster”

This variable contains the path to the Mixmaster binary on your system. It is used with various sets of parameters to gather the list of known remailers, and to finally send a message through the mixmaster chain.

**move**

Type: quadoption

Default: no

Controls whether or not Mutt will move read messages from your spool mailbox to your \$mbox mailbox, or as a result of a “mbox-hook” command.

**muttlisp\_inline\_eval**

Type: boolean

Default: no

If *set*, Mutt will evaluate bare parenthesis arguments to commands as MuttLisp expressions.

**narrow\_tree**

Type: boolean

Default: no

This variable, when *set*, makes the thread tree narrower, allowing deeper threads to fit on the screen.

**net\_inc**

Type: number

Default: 10

Operations that expect to transfer a large amount of data over the network will update their progress every \$net\_inc kilobytes. If set to 0, no progress messages will be displayed.

See also \$read\_inc, \$write\_inc and \$net\_inc.

**new\_mail\_command**

Type: path

Default: “”

If *set*, Mutt will call this command after a new message is received. See the \$status\_format documentation for the values that can be formatted into this command.

**pager**

Type: path

Default: “builtin”

This variable specifies which pager you would like to use to view messages. The value “builtin” means to use the built-in pager, otherwise this variable should specify the pathname of the external pager you would like to use.

Using an external pager may have some disadvantages: Additional keystrokes are necessary because you can’t call mutt functions directly from the pager, and screen resizes cause lines longer than the screen width to be badly formatted in the help menu.

When using an external pager, also see \$prompt\_after which defaults *set*.

**pager\_context**

Type: number

Default: 0

This variable controls the number of lines of context that are given when displaying the next or previous page in the internal pager. By default, Mutt will display the line after the last one on the screen at the top of the next page (0 lines of context).

This variable also specifies the amount of context given for search results. If positive, this many lines will be given before a match, if 0, the match will be top-aligned.

**pager\_format**

Type: string

Default: “-%Z- %C/%m: %-20.20n %s%\* — (%P)”

This variable controls the format of the one-line message “status” displayed before each message in either the internal or an external pager. The valid sequences are listed in the \$index\_format section.

**pager\_index\_lines**

Type: number

Default: 0

Determines the number of lines of a mini-index which is shown when in the pager. The current message, unless near the top or bottom of the folder, will be roughly one third of the way down this mini-index, giving the reader the context of a few messages before and after the message. This is useful, for example, to determine how many messages remain to be read in the current thread. One of the lines is reserved for the status bar from the index, so a setting of 6 will only show 5 lines of the actual index. A value of 0 results in no index being shown. If the number of messages in the current folder is less than \$pager\_index\_lines, then the index will only use as many lines as it needs.

**pager\_stop**

Type: boolean

Default: no

When *set*, the internal-pager will **not** move to the next message when you are at the end of a message and invoke the **<next-page>** function.

**pattern\_format**

Type: string

Default: “%2n %-15e %d”

This variable describes the format of the “pattern completion” menu. The following **printf(3)**-style sequences are understood:

%d      pattern description

%e      pattern expression

%n      index number

**pgp\_auto\_decode**

Type: boolean

Default: no

If *set*, mutt will automatically attempt to decrypt traditional PGP messages whenever the user performs an operation which ordinarily would result in the contents of the message being operated on. For example, if the user displays a pgp-traditional message which has not been manually checked with the **<check-traditional-pgp>** function, mutt will automatically check the message for traditional pgp.

**pgp\_autoinline**

Type: boolean

Default: no

This option controls whether Mutt generates old-style inline (traditional) PGP encrypted or signed messages under certain circumstances. This can be overridden by use of the pgp menu, when inline is not required. The GPGME backend does not support this option.

Note that Mutt might automatically use PGP/MIME for messages which consist of more than a single MIME part. Mutt can be configured to ask before sending PGP/MIME messages when inline (traditional) would not work.

Also see the \$pgp\_mime\_auto variable.

Also note that using the old-style PGP message format is **strongly deprecated**. (PGP only)

**pgp\_check\_exit**

Type: boolean

Default: yes

If *set*, mutt will check the exit code of the PGP subprocess when signing or encrypting. A non-zero exit code means that the subprocess failed. (PGP only)

#### **pgp\_check\_gpg\_decrypt\_status\_fd**

Type: boolean

Default: yes

If *set*, mutt will check the status file descriptor output of \$pgp\_decrypt\_command and \$pgp\_decode\_command for GnuPG status codes indicating successful decryption. This will check for the presence of DECRYPTION\_OKAY, absence of DECRYPTION\_FAILED, and that all PLAIN-TEXT occurs between the BEGIN\_DECRYPTION and END\_DECRYPTION status codes.

If *unset*, mutt will instead match the status fd output against \$pgp\_decryption\_okay. (PGP only)

#### **pgp\_clearsign\_command**

Type: string

Default: ""

This format is used to create an old-style "clearsigned" PGP message. Note that the use of this format is **strongly deprecated**.

This is a format string, see the \$pgp\_decode\_command command for possible **printf(3)**-like sequences. (PGP only)

#### **pgp\_decode\_command**

Type: string

Default: ""

This format string specifies a command which is used to decode application/pgp attachments.

The PGP command formats have their own set of **printf(3)**-like sequences:

- %p Expands to PGPPASSFD=0 when a pass phrase is needed, to an empty string otherwise.  
Note: This may be used with a %? construct.
- %f Expands to the name of a file containing a message.
- %s Expands to the name of a file containing the signature part  
of a **multipart/signed** attachment when verifying it.
- %a The value of \$pgp\_sign\_as if set, otherwise the value of \$pgp\_default\_key.
- %r One or more key IDs (or fingerprints if available).

For examples on how to configure these formats for the various versions of PGP which are floating around, see the pgp and gpg sample configuration files in the **samples/** subdirectory which has been installed on your system alongside the documentation. (PGP only)

#### **pgp\_decrypt\_command**

Type: string

Default: ""



This command is used to decrypt a PGP encrypted message.

This is a format string, see the `$pgp_decode_command` command for possible **printf(3)**-like sequences. (PGP only)

#### **pgp\_decryption\_okay**

Type: regular expression

Default: ""

If you assign text to this variable, then an encrypted PGP message is only considered successfully decrypted if the output from `$pgp_decrypt_command` contains the text. This is used to protect against a spoofed encrypted message, with multipart/encrypted headers but containing a block that is not actually encrypted. (e.g. simply signed and ascii armored text).

Note that if `$pgp_check_gpg_decrypt_status_fd` is set, this variable is ignored. (PGP only)

#### **pgp\_default\_key**

Type: string

Default: ""

This is the default key-pair to use for PGP operations. It will be used for encryption (see `$postpone_encrypt` and `$pgp_self_encrypt`).

It will also be used for signing unless `$pgp_sign_as` is set.

The (now deprecated) `pgp_self_encrypt_as` is an alias for this variable, and should no longer be used. (PGP only)

#### **pgp\_encrypt\_only\_command**

Type: string

Default: ""

This command is used to encrypt a body part without signing it.

This is a format string, see the `$pgp_decode_command` command for possible **printf(3)**-like sequences. (PGP only)

#### **pgp\_encrypt\_sign\_command**

Type: string

Default: ""

This command is used to both sign and encrypt a body part.

This is a format string, see the `$pgp_decode_command` command for possible **printf(3)**-like sequences. (PGP only)

**pgp\_entry\_format**

Type: string

Default: “%4n %t%f %4l/0x%k %-4a %2c %u”

This variable allows you to customize the PGP key selection menu to your personal taste. This string is similar to \$index\_format, but has its own set of **printf(3)**–like sequences:

%n	number
%k	key id
%u	user id
%a	algorithm
%l	key length
%f	flags
%c	capabilities
%t	trust/validity of the key–uid association
%[<s>]	date of the key where <s> is an <b>strftime(3)</b> expression

(PGP only)

**pgp\_export\_command**

Type: string

Default: “”

This command is used to export a public key from the user’s key ring.

This is a format string, see the \$pgp\_decode\_command command for possible **printf(3)**–like sequences. (PGP only)

**pgp\_getkeys\_command**

Type: string

Default: “”

This command is invoked whenever Mutt needs to fetch the public key associated with an email address. Of the sequences supported by \$pgp\_decode\_command, %r is the only **printf(3)**–like sequence used with this format. Note that in this case, %r expands to the email address, not the public key ID (the key ID is unknown, which is why Mutt is invoking this command). (PGP only)

**pgp\_good\_sign**

Type: regular expression

Default: “”

If you assign a text to this variable, then a PGP signature is only considered verified if the output from \$pgp\_verify\_command contains the text. Use this variable if the exit code from the command is 0 even for bad signatures. (PGP only)

**pgp\_ignore\_subkeys**

Type: boolean  
Default: yes

Setting this variable will cause Mutt to ignore OpenPGP subkeys. Instead, the principal key will inherit the subkeys' capabilities. *Unset* this if you want to play interesting key selection games. (PGP only)

**pgp\_import\_command**

Type: string  
Default: ""

This command is used to import a key from a message into the user's public key ring.

This is a format string, see the `$pgp_decode_command` command for possible **printf(3)**-like sequences. (PGP only)

**pgp\_list\_pubring\_command**

Type: string  
Default: ""

This command is used to list the public key ring's contents. The output format must be analogous to the one used by

```
gpg --list-keys --with-colons --with-fingerprint
```

This format is also generated by the **mutt\_pgpring** utility which comes with mutt.

Note: gpg's **fixed-list-mode** option should not be used. It produces a different date format which may result in mutt showing incorrect key generation dates.

This is a format string, see the `$pgp_decode_command` command for possible **printf(3)**-like sequences. Note that in this case, `%r` expands to the search string, which is a list of one or more quoted values such as email address, name, or keyid. (PGP only)

**pgp\_list\_secring\_command**

Type: string  
Default: ""

This command is used to list the secret key ring's contents. The output format must be analogous to the one used by:

```
gpg --list-keys --with-colons --with-fingerprint
```

This format is also generated by the **mutt\_pgpring** utility which comes with mutt.

Note: gpg's **fixed-list-mode** option should not be used. It produces a different date format which may result in mutt showing incorrect key generation dates.

This is a format string, see the `$pgp_decode_command` command for possible **printf(3)**-like sequences. Note that in this case, `%r` expands to the search string, which is a list of one or more quoted values such as email address, name, or keyid. (PGP only)

### **pgp\_long\_ids**

Type: boolean

Default: yes

If *set*, use 64 bit PGP key IDs, if *unset* use the normal 32 bit key IDs. NOTE: Internally, Mutt has transitioned to using fingerprints (or long key IDs as a fallback). This option now only controls the display of key IDs in the key selection menu and a few other places. (PGP only)

### **pgp\_mime\_auto**

Type: quadoption

Default: ask-yes

This option controls whether Mutt will prompt you for automatically sending a (signed/encrypted) message using PGP/MIME when inline (traditional) fails (for any reason).

Also note that using the old-style PGP message format is **strongly deprecated**. (PGP only)

### **pgp\_replyinline**

Type: boolean

Default: no

Setting this variable will cause Mutt to always attempt to create an inline (traditional) message when replying to a message which is PGP encrypted/signed inline. This can be overridden by use of the `pgp` menu, when inline is not required. This option does not automatically detect if the (replied-to) message is inline; instead it relies on Mutt internals for previously checked/flagged messages.

Note that Mutt might automatically use PGP/MIME for messages which consist of more than a single MIME part. Mutt can be configured to ask before sending PGP/MIME messages when inline (traditional) would not work.

Also see the `$pgp_mime_auto` variable.

Also note that using the old-style PGP message format is **strongly deprecated**. (PGP only)

**pgp\_retainable\_sigs**

Type: boolean

Default: no

If *set*, signed and encrypted messages will consist of nested **multipart/signed** and **multipart/encrypted** body parts.

This is useful for applications like encrypted and signed mailing lists, where the outer layer (**multipart/encrypted**) can be easily removed, while the inner **multipart/signed** part is retained. (PGP only)

**pgp\_self\_encrypt**

Type: boolean

Default: yes

When *set*, PGP encrypted messages will also be encrypted using the key in `$pgp_default_key`. (PGP only)

**pgp\_show\_unusable**

Type: boolean

Default: yes

If *set*, mutt will display non–usable keys on the PGP key selection menu. This includes keys which have been revoked, have expired, or have been marked as “disabled” by the user. (PGP only)

**pgp\_sign\_as**

Type: string

Default: “”

If you have a different key pair to use for signing, you should set this to the signing key. Most people will only need to set `$pgp_default_key`. It is recommended that you use the keyid form to specify your key (e.g. **0x00112233**). (PGP only)

**pgp\_sign\_command**

Type: string

Default: “”

This command is used to create the detached PGP signature for a **multipart/signed** PGP/MIME body part.

This is a format string, see the `$pgp_decode_command` command for possible **printf(3)**–like sequences. (PGP only)

**pgp\_sort\_keys**

Type: sort order  
Default: address

Specifies how the entries in the `pgp` menu are sorted. The following are legal values:

<code>address</code>	sort alphabetically by user id
<code>keyid</code>	sort alphabetically by key id
<code>date</code>	sort by key creation date
<code>trust</code>	sort by the trust of the key

If you prefer reverse order of the above values, prefix it with “reverse-”. (PGP only)

**pgp\_strict\_enc**

Type: boolean  
Default: yes

If *set*, Mutt will automatically encode PGP/MIME signed messages as quoted-printable. Please note that unsetting this variable may lead to problems with non-verifyable PGP signatures, so only change this if you know what you are doing. (PGP only)

**pgp\_timeout**

Type: number (long)  
Default: 300

The number of seconds after which a cached passphrase will expire if not used. (PGP only)

**pgp\_use\_gpg\_agent**

Type: boolean  
Default: yes

If *set*, mutt expects a **gpg-agent(1)** process will handle private key passphrase prompts. If *unset*, mutt will prompt for the passphrase and pass it via stdin to the `pgp` command.

Note that as of version 2.1, GnuPG automatically spawns an agent and requires the agent be used for passphrase management. Since that version is increasingly prevalent, this variable now defaults *set*.

Mutt works with a GUI or curses pinentry program. A TTY pinentry should not be used.

If you are using an older version of GnuPG without an agent running, or another encryption program without an agent, you will need to *unset* this variable. (PGP only)

**pgp\_verify\_command**

Type: string  
Default: “”

This command is used to verify PGP signatures.

This is a format string, see the `$pgp_decode_command` command for possible **printf(3)**-like sequences. (PGP only)

#### **pgp\_verify\_key\_command**

Type: string

Default: ""

This command is used to verify key information from the key selection menu.

This is a format string, see the `$pgp_decode_command` command for possible **printf(3)**-like sequences. (PGP only)

#### **pipe\_decode**

Type: boolean

Default: no

Used in connection with the `<pipe-message>` function. When *unset*, Mutt will pipe the messages without any preprocessing. When *set*, Mutt will attempt to decode the messages first.

Also see `$pipe_decode_weed`, which controls whether headers will be weeded when this is *set*.

#### **pipe\_decode\_weed**

Type: boolean

Default: yes

For `<pipe-message>`, when `$pipe_decode` is set, this further controls whether Mutt will weed headers.

#### **pipe\_sep**

Type: string

Default: "\n"

The separator to add between messages when piping a list of tagged messages to an external Unix command.

#### **pipe\_split**

Type: boolean

Default: no

Used in connection with the `<pipe-message>` function following `<tag-prefix>`. If this variable is *unset*, when piping a list of tagged messages Mutt will concatenate the messages and will pipe them all concatenated. When *set*, Mutt will pipe the messages one by one. In both cases the

messages are piped in the current sorted order, and the `$pipe_sep` separator is added after each message.

**pop\_auth\_try\_all**

Type: boolean

Default: yes

If *set*, Mutt will try all available authentication methods. When *unset*, Mutt will only fall back to other authentication methods if the previous methods are unavailable. If a method is available but authentication fails, Mutt will not connect to the POP server.

**pop\_authenticators**

Type: string

Default: ""

This is a colon-delimited list of authentication methods mutt may attempt to use to log in to an POP server, in the order mutt should try them. Authentication methods are either "user", "apop" or any SASL mechanism, e.g. "digest-md5", "gssapi" or "cram-md5". This option is case-insensitive. If this option is *unset* (the default) mutt will try all available methods, in order from most-secure to least-secure.

Example:

```
set pop_authenticators="digest-md5:apop:user"
```

**pop\_checkinterval**

Type: number

Default: 60

This variable configures how often (in seconds) mutt should look for new mail in the currently selected mailbox if it is a POP mailbox.

**pop\_delete**

Type: quadoption

Default: ask-no

If *set*, Mutt will delete successfully downloaded messages from the POP server when using the `<fetch-mail>` function. When *unset*, Mutt will download messages but also leave them on the POP server.



**pop\_host**

Type: string

Default: ""

The name of your POP server for the **<fetch-mail>** function. You can also specify an alternative port, username and password, i.e.:

```
[pop[s]://][username[:password]@]popserver[:port]
```

where “[...]” denotes an optional part.

**pop\_last**

Type: boolean

Default: no

If this variable is *set*, mutt will try to use the “**LAST**” POP command for retrieving only unread messages from the POP server when using the **<fetch-mail>** function.

**pop\_oauth\_refresh\_command**

Type: string

Default: ""

The command to run to generate an OAUTH refresh token for authorizing your connection to your POP server. This command will be run on every connection attempt that uses the OAUTH-BEARER authentication mechanism. See “oauth” for details.

**pop\_pass**

Type: string

Default: ""

Specifies the password for your POP account. If *unset*, Mutt will prompt you for your password when you open a POP mailbox.

**Warning:** you should only use this option when you are on a fairly secure machine, because the superuser can read your muttrc even if you are the only one who can read the file.

**pop\_reconnect**

Type: quadoption

Default: ask-yes

Controls whether or not Mutt will try to reconnect to the POP server if the connection is lost.

**pop\_user**

Type: string  
Default: ""

Your login name on the POP server.

This variable defaults to your user name on the local machine.

**post\_indent\_string**

Type: string  
Default: ""

Similar to the \$attribution variable, Mutt will append this string after the inclusion of a message which is being replied to. For a full listing of defined **printf(3)**-like sequences see the section on \$index\_format.

**postpone**

Type: quadoption  
Default: ask=yes

Controls whether or not messages are saved in the \$postponed mailbox when you elect not to send immediately.

Also see the \$recall variable.

**postponed**

Type: path  
Default: "~/postponed"

Mutt allows you to indefinitely "postpone sending a message" which you are editing. When you choose to postpone a message, Mutt saves it in the mailbox specified by this variable.

Also see the \$postpone variable.

**postpone\_encrypt**

Type: boolean  
Default: no

When *set*, postponed messages that are marked for encryption will be self-encrypted. Mutt will first try to encrypt using the value specified in \$pgp\_default\_key or \$smime\_default\_key. If those are not set, it will try the deprecated \$postpone\_encrypt\_as. (Crypto only)

**postpone\_encrypt\_as**

Type: string  
Default: ""

This is a deprecated fall-back variable for \$postpone\_encrypt. Please use \$pgp\_default\_key or \$smime\_default\_key. (Crypto only)

**preconnect**

Type: string  
Default: ""

If *set*, a shell command to be executed if mutt fails to establish a connection to the server. This is useful for setting up secure connections, e.g. with **ssh(1)**. If the command returns a nonzero status, mutt gives up opening the server. Example:

```
set preconnect="ssh -f -q -L 1234:mailhost.net:143 mailhost.net \  
sleep 20 < /dev/null > /dev/null"
```

Mailbox "foo" on "mailhost.net" can now be reached as "{localhost:1234}foo".

Note: For this example to work, you must be able to log in to the remote machine without having to enter a password.

**print**

Type: quadoption  
Default: ask-no

Controls whether or not Mutt really prints messages. This is set to "ask-no" by default, because some people accidentally hit "p" often.

**print\_command**

Type: path  
Default: "lpr"

This specifies the command pipe that should be used to print messages.

**print\_decode**

Type: boolean  
Default: yes

Used in connection with the **<print-message>** function. If this option is *set*, the message is decoded before it is passed to the external command specified by \$print\_command. If this option is *unset*, no processing will be applied to the message when printing it. The latter setting may be useful if you are using some advanced printer filter which is able to properly format e-mail

messages for printing.

Also see `$print_decode_weed`, which controls whether headers will be weeded when this is *set*.

#### **print\_decode\_weed**

Type: boolean

Default: yes

For `<print-message>`, when `$print_decode` is set, this further controls whether Mutt will weed headers.

#### **print\_split**

Type: boolean

Default: no

Used in connection with the `<print-message>` function. If this option is *set*, the command specified by `$print_command` is executed once for each message which is to be printed. If this option is *unset*, the command specified by `$print_command` is executed only once, and all the messages are concatenated, with a form feed as the message separator.

Those who use the **enscript**(1) program's mail-printing mode will most likely want to *set* this option.

#### **prompt\_after**

Type: boolean

Default: yes

If you use an *external* `$pager`, setting this variable will cause Mutt to prompt you for a command when the pager exits rather than returning to the index menu. If *unset*, Mutt will return to the index menu when the external pager exits.

#### **query\_command**

Type: path

Default: ""

This specifies the command Mutt will use to make external address queries. The string may contain a `"%s"`, which will be substituted with the query string the user types. Mutt will add quotes around the string substituted for `"%s"` automatically according to shell quoting rules, so you should avoid adding your own. If no `"%s"` is found in the string, Mutt will append the user's query to the end of the string. See "query" for more information.

#### **query\_format**

Type: string

Default: `"%4c %t %-25.25a %-25.25n %?e?(%e)?"`

This variable describes the format of the “query” menu. The following **printf(3)**–style sequences are understood:

%a	destination address
%c	current entry number
%e	extra information *
%n	destination name
%t	“*” if current entry is tagged, a space otherwise
%>X	right justify the rest of the string and pad with “X”
% X	pad to the end of the line with “X”
%*X	soft-fill with character “X” as pad

For an explanation of “soft-fill”, see the `$index_format` documentation.

\* = can be optionally printed if nonzero, see the `$status_format` documentation.

## quit

Type: quadoption  
Default: yes

This variable controls whether “quit” and “exit” actually quit from mutt. If this option is *isset*, the y do quit, if it is *unset*, they have no effect, and if it is set to *ask=yes* or *ask=no*, you are prompted for confirmation when you try to quit.

## quote\_regexp

Type: regular expression  
Default: “`^([\t]*[>:}#])+`”

A regular expression used in the internal pager to determine quoted sections of text in the body of a message. Quoted text may be filtered out using the `<toggle-quoted>` command, or colored according to the “color quoted” family of directives.

Higher levels of quoting may be colored differently (“color quoted1”, “color quoted2”, etc.). The quoting level is determined by removing the last character from the matched text and recursively reapplying the regular expression until it fails to produce a match.

Match detection may be overridden by the `$smileys` regular expression.

## read\_inc

Type: number  
Default: 10

If set to a value greater than 0, Mutt will display which message it is currently on when reading a mailbox or when performing search actions such as search and limit. The message is printed after this many messages have been read or searched (e.g., if set to 25, Mutt will print a message when it is at message 25, and then again when it gets to message 50). This variable is meant to indicate progress when reading or searching large mailboxes which may take some time. When set to 0, only a single message will appear before the reading the mailbox.

Also see the `$write_inc`, `$net_inc` and `$time_inc` variables and the “tuning” section of the manual for performance considerations.

**read\_only**

Type: boolean

Default: no

If *set*, all folders are opened in read-only mode.

**realname**

Type: string

Default: “”

This variable specifies what “real” or “personal” name should be used when sending messages.

By default, this is the GECOS field from `/etc/passwd`. Note that this variable will *not* be used when the user has set a real name in the `$from` variable.

**recall**

Type: quadoption

Default: ask-yes

Controls whether or not Mutt recalls postponed messages when composing a new message.

Setting this variable to *yes* is not generally useful, and thus not recommended. Note that the `<recall-message>` function can be used to manually recall postponed messages.

Also see `$postponed` variable.

**record**

Type: path

Default: “~/sent”

This specifies the file into which your outgoing messages should be appended. (This is meant as the primary method for saving a copy of your messages, but another way to do this is using the “my\_hdr” command to create a “Bcc:” field with your email address in it.)

The value of `$record` is overridden by the `$force_name` and `$save_name` variables, and the “fcc-hook” command. Also see `$copy` and `$write_bcc`.

Multiple mailboxes may be specified if `$fcc_delimiter` is set to a string delimiter.

**reflow\_space\_quotes**

Type: boolean

Default: yes

This option controls how quotes from `format=flowed` messages are displayed in the pager and when replying (with `$text_flowed unset`). When set, this option adds spaces after each level of quote marks, turning `">>>foo"` into `"> > > foo"`.

**Note:** If `$reflow_text` is *unset*, this option has no effect. Also, this option does not affect replies when `$text_flowed` is *set*.

**reflow\_text**

Type: boolean

Default: yes

When *set*, Mutt will reformat paragraphs in text/plain parts marked `format=flowed`. If *unset*, Mutt will display paragraphs unaltered from how they appear in the message body. See RFC3676 for details on the *format=flowed* format.

Also see `$reflow_wrap`, and `$wrap`.

**reflow\_wrap**

Type: number

Default: 78

This variable controls the maximum paragraph width when reformatting text/plain parts when `$reflow_text` is *set*. When the value is 0, paragraphs will be wrapped at the terminal's right margin. A positive value sets the paragraph width relative to the left margin. A negative value set the paragraph width relative to the right margin.

Also see `$wrap`.

**reply\_regexp**

Type: regular expression

Default: `“^(re([0-9\+])*\|aw):[ \t]*”`

A regular expression used to recognize reply messages when threading and replying. The default value corresponds to the English `Re:` and the German `Aw:`.

**reply\_self**

Type: boolean

Default: no

If *unset* and you are replying to a message sent by you, Mutt will assume that you want to reply to the recipients of that message rather than to yourself.

Also see the “alternates” command.

**reply\_to**

Type: quadoption

Default: ask=yes

If *set*, when replying to a message, Mutt will use the address listed in the Reply-to: header as the recipient of the reply. If *unset*, it will use the address in the From: header field instead. This option is useful for reading a mailing list that sets the Reply-To: header field to the list address and you want to send a private message to the author of a message.

**resolve**

Type: boolean

Default: yes

When *set*, the cursor will be automatically advanced to the next (possibly undeleted) message whenever a command that modifies the current message is executed.

**resume\_draft\_files**

Type: boolean

Default: no

If *set*, draft files (specified by **-H** on the command line) are processed similarly to when resuming a postponed message. Recipients are not prompted for; send-hooks are not evaluated; no alias expansion takes place; user-defined headers and signatures are not added to the message.

**resume\_edited\_draft\_files**

Type: boolean

Default: yes

If *set*, draft files previously edited (via **-E -H** on the command line) will have \$resume\_draft\_files automatically set when they are used as a draft file again.

The first time a draft file is saved, mutt will add a header, X-Mutt-Resume-Draft to the saved file. The next time the draft file is read in, if mutt sees the header, it will set \$resume\_draft\_files.

This option is designed to prevent multiple signatures, user-defined headers, and other processing effects from being made multiple times to the draft file.

**reverse\_alias**

Type: boolean

Default: no



This variable controls whether or not Mutt will display the “personal” name from your aliases in the index menu if it finds an alias that matches the message’s sender. For example, if you have the following alias:

```
alias juser abd30425@somewhere.net (Joe User)
```

and then you receive mail which contains the following header:

```
From: abd30425@somewhere.net
```

It would be displayed in the index menu as “Joe User” instead of “abd30425@somewhere.net.” This is useful when the person’s e-mail address is not human friendly.

#### **reverse\_name**

Type: boolean

Default: no

It may sometimes arrive that you receive mail to a certain machine, move the messages to another machine, and reply to some the messages from there. If this variable is *set*, the default *From:* line of the reply messages is built using the address where you received the messages you are replying to *if* that address matches your “alternates”. If the variable is *unset*, or the address that would be used doesn’t match your “alternates”, the *From:* line will use your address on the current machine.

Also see the “alternates” command and `$reverse_realname`.

#### **reverse\_realname**

Type: boolean

Default: yes

This variable fine-tunes the behavior of the `$reverse_name` feature.

When it is *unset*, Mutt will remove the real name part of a matching address. This allows the use of the email address without having to also use what the sender put in the real name field.

When it is *set*, Mutt will use the matching address as-is.

In either case, a missing real name will be filled in afterwards using the value of `$realname`.

#### **rfc2047\_parameters**

Type: boolean

Default: no

When this variable is *set*, Mutt will decode RFC2047-encoded MIME parameters. You want to set this variable when mutt suggests you to save attachments to files named like:

=?iso-8859-1?Q?file=5F=E4=5F991116=2Ezip?=

When this variable is *set* interactively, the change won't be active until you change folders.

Note that this use of RFC2047's encoding is explicitly prohibited by the standard, but nevertheless encountered in the wild.

Also note that setting this parameter will *not* have the effect that mutt *generates* this kind of encoding. Instead, mutt will unconditionally use the encoding specified in RFC2231.

#### **save\_address**

Type: boolean

Default: no

If *set*, mutt will take the sender's full address when choosing a default folder for saving a mail. If \$save\_name or \$force\_name is *set* too, the selection of the Fcc folder will be changed as well.

#### **save\_empty**

Type: boolean

Default: yes

When *unset*, mailboxes which contain no saved messages will be removed when closed (the exception is \$spoolfile which is never removed). If *set*, mailboxes are never removed.

**Note:** This only applies to mbox and MMDF folders, Mutt does not delete MH and Maildir directories.

#### **save\_history**

Type: number

Default: 0

This variable controls the size of the history (per category) saved in the \$history\_file file.

#### **save\_name**

Type: boolean

Default: no

This variable controls how copies of outgoing messages are saved. When *set*, a check is made to see if a mailbox specified by the recipient address exists (this is done by searching for a mailbox in the \$folder directory with the *username* part of the recipient address). If the mailbox exists, the outgoing message will be saved to that mailbox, otherwise the message is saved to the \$record mailbox.

Also see the \$force\_name variable.

**score**

Type: boolean  
Default: yes

When this variable is *unset*, scoring is turned off. This can be useful to selectively disable scoring for certain folders when the `$score_threshold_delete` variable and related are used.

**score\_threshold\_delete**

Type: number  
Default: -1

Messages which have been assigned a score equal to or lower than the value of this variable are automatically marked for deletion by mutt. Since mutt scores are always greater than or equal to zero, the default setting of this variable will never mark a message for deletion.

**score\_threshold\_flag**

Type: number  
Default: 9999

Messages which have been assigned a score greater than or equal to this variable's value are automatically marked "flagged".

**score\_threshold\_read**

Type: number  
Default: -1

Messages which have been assigned a score equal to or lower than the value of this variable are automatically marked as read by mutt. Since mutt scores are always greater than or equal to zero, the default setting of this variable will never mark a message read.

**search\_context**

Type: number  
Default: 0

For the pager, this variable specifies the number of lines shown before search results. By default, search results will be top-aligned.

**send\_charset**

Type: string  
Default: "us-ascii:iso-8859-1:utf-8"

A colon-delimited list of character sets for outgoing messages. Mutt will use the first character set into which the text can be converted exactly. If your `$charset` is not "iso-8859-1" and recipients may not understand "UTF-8", it is advisable to include in the list an appropriate widely used

standard character set (such as “iso-8859-2”, “koi8-r” or “iso-2022-jp”) either instead of or after “iso-8859-1”.

In case the text cannot be converted into one of these exactly, mutt uses \$charset as a fallback.

### **send\_multipart\_alternative**

Type: quadoption

Default: no

If *set*, Mutt will generate a multipart/alternative container and an alternative part using the filter script specified in \$send\_multipart\_alternative\_filter. See the section “MIME Multipart/Alternative” (alternative-order).

Note that enabling multipart/alternative is not compatible with inline PGP encryption. Mutt will prompt to use PGP/MIME in that case.

### **send\_multipart\_alternative\_filter**

Type: path

Default: “”

This specifies a filter script, which will convert the main (composed) message of the email to an alternative format. The message will be piped to the filter’s stdin. The expected output of the filter is the generated mime type, e.g. text/html, followed by a blank line, and then the converted content. See the section “MIME Multipart/Alternative” (alternative-order).

### **sendmail**

Type: path

Default: “/usr/sbin/sendmail -oem -oi”

Specifies the program and arguments used to deliver mail sent by Mutt. Mutt expects that the specified program interprets additional arguments as recipient addresses. Mutt appends all recipients after adding a — delimiter (if not already present). Additional flags, such as for \$use\_8bitmime, \$use\_envelope\_from, \$dsn\_notify, or \$dsn\_return will be added before the delimiter.

**See also:** \$write\_bcc.

### **sendmail\_wait**

Type: number

Default: 0

Specifies the number of seconds to wait for the \$sendmail process to finish before giving up and putting delivery in the background.

Mutt interprets the value of this variable as follows:

>0      number of seconds to wait for sendmail to finish before continuing

0       wait forever for sendmail to finish  
<0      always put sendmail in the background without waiting

Note that if you specify a value other than 0, the output of the child process will be put in a temporary file. If there is some error, you will be informed as to where to find the output.

### **shell**

Type: path  
Default: ""

Command to use when spawning a subshell. By default, the user's login shell from **/etc/passwd** is used.

### **sidebar\_delim\_chars**

Type: string  
Default: "/"

This contains the list of characters which you would like to treat as folder separators for displaying paths in the sidebar.

Local mail is often arranged in directories: 'dir1/dir2/mailbox'.

```
set sidebar_delim_chars='/'
```

IMAP mailboxes are often named: 'folder1.folder2.mailbox'.

```
set sidebar_delim_chars='.'
```

**See also:** \$sidebar\_short\_path, \$sidebar\_folder\_indent, \$sidebar\_indent\_string.

### **sidebar\_divider\_char**

Type: string  
Default: "|"

This specifies the characters to be drawn between the sidebar (when visible) and the other Mutt panels. ASCII and Unicode line-drawing characters are supported.

### **sidebar\_folder\_indent**

Type: boolean  
Default: no

Set this to indent mailboxes in the sidebar.

**See also:** \$sidebar\_short\_path, \$sidebar\_indent\_string, \$sidebar\_delim\_chars.

### sidebar\_format

Type: string

Default: “%B%\* %n”

This variable allows you to customize the sidebar display. This string is similar to \$index\_format, but has its own set of **printf(3)**–like sequences:

%B	Name of the mailbox
%S	* Size of mailbox (total number of messages)
%N	* Number of unread messages in the mailbox
%n	N if mailbox has new mail, blank otherwise
%F	* Number of Flagged messages in the mailbox
%!	“!” : one flagged message; “!!” : two flagged messages; “n!” : n flagged messages (for n > 2). Otherwise prints nothing.
%d	* @ Number of deleted messages
%L	* @ Number of messages after limiting
%t	* @ Number of tagged messages
%>X	right justify the rest of the string and pad with “X”
% X	pad to the end of the line with “X”
%%X	soft-fill with character “X” as pad

\* = Can be optionally printed if nonzero @ = Only applicable to the current folder

In order to use %S, %N, %F, and %!, \$mail\_check\_stats must be *set*. When thus set, a suggested value for this option is “%B%?F? [%F]?%\* %?N?%N/?%S”.

### sidebar\_indent\_string

Type: string

Default: “ ”

This specifies the string that is used to indent mailboxes in the sidebar. It defaults to two spaces.

**See also:** \$sidebar\_short\_path, \$sidebar\_folder\_indent, \$sidebar\_delim\_chars.

### sidebar\_new\_mail\_only

Type: boolean

Default: no

When set, the sidebar will only display mailboxes containing new, or flagged, mail.

**See also:** sidebar\_whitelist.

**sidebar\_next\_new\_wrap**

Type: boolean

Default: no

When set, the **<sidebar-next-new>** command will not stop at the end of the list of mailboxes, but wrap around to the beginning. The **<sidebar-prev-new>** command is similarly affected, wrapping around to the end of the list.

**sidebar\_relative\_shortpath\_indent**

Type: boolean

Default: no

When set, this option changes how `$sidebar_short_path` and `$sidebar_folder_indent` perform shortening and indentation: both will look at the previous sidebar entries and shorten/indent relative to the most recent parent.

An example of this option set/unset for mailboxes listed in this order, with `$sidebar_short_path=yes`, `$sidebar_folder_indent=yes`, and `$sidebar_indent_string="→"`:

**mailbox**

	set	unset
<b>=a.b</b>	<b>=a.b →b</b>	
<b>=a.b.c.d</b>		
	<b>→c.d →→→d</b>	
<b>=a.b.e</b>	<b>→e →→→e</b>	

The second line illustrates most clearly. With this option set, **=a.b.c.d** is shortened relative to **=a.b**, becoming **c.d**; it is also indented one place relative to **=a.b**. With this option unset **=a.b.c.d** is always shortened to the last part of the mailbox, **d** and is indented three places, with respect to `$folder` (represented by `'=`').

When set, the third line will also be indented and shortened relative to the first line.

**sidebar\_short\_path**

Type: boolean

Default: no

By default the sidebar will show the mailbox's path, relative to the `$folder` variable. Setting **sidebar\_shortpath=yes** will shorten the names relative to the previous name. Here's an example:

	shortpath=no	shortpath=yes shortpath=yes, folderindent=yes, indentstr=".."
<b>fruit</b>	<b>fruit fruit</b>	
<b>fruit.apple</b>		<b>apple ..apple</b>
<b>fruit.banana</b>		<b>banana ..banana</b>
<b>fruit.cherry</b>		<b>cherry ..cherry</b>

See also: `$sidebar_delim_chars`, `$sidebar_folder_indent`, `$sidebar_indent_string`.

**sidebar\_sort\_method**

Type: sort order

Default: unsorted

Specifies how to sort mailbox entries in the sidebar. By default, the entries are sorted alphabetically. Valid values:

- alpha (alphabetically)
- count (all message count)
- flagged (flagged message count)
- name (alphabetically)
- new (unread message count)
- path (alphabetically)
- unread (unread message count)
- unsorted

You may optionally use the “reverse-” prefix to specify reverse sorting order (example: “**set sidebar\_sort\_method=reverse-alpha**”).

**sidebar\_use\_mailbox\_shortcuts**

Type: boolean

Default: no

When set, sidebar mailboxes will be displayed with mailbox shortcut prefixes “=” or “~”.

When unset, the sidebar will trim off a matching \$folder prefix but otherwise not use mailbox shortcuts.

**sidebar\_visible**

Type: boolean

Default: no

This specifies whether or not to show sidebar. The sidebar shows a list of all your mailboxes.

**See also:** \$sidebar\_format, \$sidebar\_width

**sidebar\_width**

Type: number

Default: 30

This controls the width of the sidebar. It is measured in screen columns. For example: sidebar\_width=20 could display 20 ASCII characters, or 10 Chinese characters.

**sig\_dashes**

Type: boolean

Default: yes



If *set*, a line containing “— ” (note the trailing space) will be inserted before your \$signature. It is **strongly** recommended that you not *unset* this variable unless your signature contains just your name. The reason for this is because many software packages use “— \n” to detect your signature. For example, Mutt has the ability to highlight the signature in a different color in the built-in pager.

**sig\_on\_top**

Type: boolean  
Default: no

If *set*, the signature will be included before any quoted or forwarded text. It is **strongly** recommended that you do not set this variable unless you really know what you are doing, and are prepared to take some heat from netiquette guardians.

**signature**

Type: path  
Default: “~/signature”

Specifies the filename of your signature, which is appended to all outgoing messages. If the filename ends with a pipe (“|”), it is assumed that filename is a shell command and input should be read from its standard output.

**simple\_search**

Type: string  
Default: “~f %s | ~s %s”

Specifies how Mutt should expand a simple search into a real search pattern. A simple search is one that does not contain any of the “” pattern modifiers. See “patterns” for more information on search patterns.

For example, if you simply type “joe” at a search or limit prompt, Mutt will automatically expand it to the value specified by this variable by replacing “%s” with the supplied string. For the default value, “joe” would be expanded to: “~f joe | ~s joe”.

**size\_show\_bytes**

Type: boolean  
Default: no

If *set*, message sizes will display bytes for values less than 1 kilobyte. See `formatstrings-size`.

**size\_show\_fractions**

Type: boolean  
Default: yes

If *set*, message sizes will be displayed with a single decimal value for sizes from 0 to 10 kilobytes and 1 to 10 megabytes. See `formatstrings-size`.

**size\_show\_mb**

Type: boolean

Default: yes

If *set*, message sizes will display megabytes for values greater than or equal to 1 megabyte. See `formatstrings-size`.

**size\_units\_on\_left**

Type: boolean

Default: no

If *set*, message sizes units will be displayed to the left of the number. See `formatstrings-size`.

**sleep\_time**

Type: number

Default: 1

Specifies time, in seconds, to pause while displaying certain informational messages, while moving from folder to folder and after expunging messages from the current folder. The default is to pause one second, so a value of zero for this option suppresses the pause.

**smart\_wrap**

Type: boolean

Default: yes

Controls the display of lines longer than the screen width in the internal pager. If *set*, long lines are wrapped at a word boundary. If *unset*, lines are simply wrapped at the screen edge. Also see the `$markers` variable.

**smileys**

Type: regular expression

Default: “(>From )|(:[-^]?[!@]><){/DP)”

The *pager* uses this variable to catch some common false positives of `$quote_regexp`, most notably smileys and not consider a line quoted text if it also matches `$smileys`. This mostly happens at the beginning of a line.

**smime\_ask\_cert\_label**

Type: boolean  
Default: yes

This flag controls whether you want to be asked to enter a label for a certificate about to be added to the database or not. It is *set* by default. (S/MIME only)

**smime\_ca\_location**

Type: path  
Default: ""

This variable contains the name of either a directory, or a file which contains trusted certificates for use with OpenSSL. (S/MIME only)

**smime\_certificates**

Type: path  
Default: ""

Since for S/MIME there is no pubring/secring as with PGP, mutt has to handle storage and retrieval of keys by itself. This is very basic right now, and keys and certificates are stored in two different directories, both named as the hash-value retrieved from OpenSSL. There is an index file which contains mailbox-address keyid pairs, and which can be manually edited. This option points to the location of the certificates. (S/MIME only)

**smime\_decrypt\_command**

Type: string  
Default: ""

This format string specifies a command which is used to decrypt **application/x-pkcs7-mime** attachments.

The OpenSSL command formats have their own set of **printf(3)**-like sequences similar to PGP's:

%f	Expands to the name of a file containing a message.
%s	Expands to the name of a file containing the signature part of a <b>multipart/signed</b> attachment when verifying it.
%k	The key-pair specified with \$smime_default_key
%c	One or more certificate IDs.
%a	The algorithm used for encryption.
%d	The message digest algorithm specified with \$smime_sign_digest_alg.
%C	CA location: Depending on whether \$smime_ca_location points to a directory or file, this expands to "--CApath \$smime_ca_location" or "--CAfile \$smime_ca_location".

For examples on how to configure these formats, see the **smime.rc** in the **samples/** subdirectory which has been installed on your system alongside the documentation. (S/MIME only)

**smime\_decrypt\_use\_default\_key**

Type: boolean

Default: yes

If *set* (default) this tells mutt to use the default key for decryption. Otherwise, if managing multiple certificate–key–pairs, mutt will try to use the mailbox–address to determine the key to use. It will ask you to supply a key, if it can't find one. (S/MIME only)

**smime\_default\_key**

Type: string

Default: ""

This is the default key–pair to use for S/MIME operations, and must be set to the keyid (the hash–value that OpenSSL generates) to work properly.

It will be used for encryption (see \$postpone\_encrypt and \$smime\_self\_encrypt). If GPGME is enabled, this is the key id displayed by gpgsm.

It will be used for decryption unless \$smime\_decrypt\_use\_default\_key is *unset*.

It will also be used for signing unless \$smime\_sign\_as is set.

The (now deprecated) *smime\_self\_encrypt\_as* is an alias for this variable, and should no longer be used. (S/MIME only)

**smime\_encrypt\_command**

Type: string

Default: ""

This command is used to create encrypted S/MIME messages.

This is a format string, see the \$smime\_decrypt\_command command for possible **printf(3)**–like sequences. (S/MIME only)

**smime\_encrypt\_with**

Type: string

Default: "aes256"

This sets the algorithm that should be used for encryption. Valid choices are "aes128", "aes192", "aes256", "des", "des3", "rc2-40", "rc2-64", "rc2-128". (S/MIME only)

**smime\_get\_cert\_command**

Type: string

Default: ""

This command is used to extract X509 certificates from a PKCS7 structure.

This is a format string, see the `$smime_decrypt_command` command for possible **printf(3)**-like sequences. (S/MIME only)

#### **smime\_get\_cert\_email\_command**

Type: string

Default: ""

This command is used to extract the mail address(es) used for storing X509 certificates, and for verification purposes (to check whether the certificate was issued for the sender's mailbox).

This is a format string, see the `$smime_decrypt_command` command for possible **printf(3)**-like sequences. (S/MIME only)

#### **smime\_get\_signer\_cert\_command**

Type: string

Default: ""

This command is used to extract only the signers X509 certificate from a S/MIME signature, so that the certificate's owner may get compared to the email's "From:" field.

This is a format string, see the `$smime_decrypt_command` command for possible **printf(3)**-like sequences. (S/MIME only)

#### **smime\_import\_cert\_command**

Type: string

Default: ""

This command is used to import a certificate via `smime_keys`.

This is a format string, see the `$smime_decrypt_command` command for possible **printf(3)**-like sequences. (S/MIME only)

#### **smime\_is\_default**

Type: boolean

Default: no

The default behavior of mutt is to use PGP on all auto-sign/encryption operations. To override and to use OpenSSL instead this must be *set*. However, this has no effect while replying, since mutt will automatically select the same application that was used to sign/encrypt the original message. (Note that this variable can be overridden by unsetting `$crypt_autosmime`.) (S/MIME only)

**smime\_keys**

Type: path  
Default: ""

Since for S/MIME there is no pubring/secring as with PGP, mutt has to handle storage and retrieval of keys/certs by itself. This is very basic right now, and stores keys and certificates in two different directories, both named as the hash-value retrieved from OpenSSL. There is an index file which contains mailbox-address keyid pair, and which can be manually edited. This option points to the location of the private keys. (S/MIME only)

**smime\_pk7out\_command**

Type: string  
Default: ""

This command is used to extract PKCS7 structures of S/MIME signatures, in order to extract the public X509 certificate(s).

This is a format string, see the \$smime\_decrypt\_command command for possible **printf(3)**-like sequences. (S/MIME only)

**smime\_self\_encrypt**

Type: boolean  
Default: yes

When *set*, S/MIME encrypted messages will also be encrypted using the certificate in \$smime\_default\_key. (S/MIME only)

**smime\_sign\_as**

Type: string  
Default: ""

If you have a separate key to use for signing, you should set this to the signing key. Most people will only need to set \$smime\_default\_key. (S/MIME only)

**smime\_sign\_command**

Type: string  
Default: ""

This command is used to create S/MIME signatures of type **multipart/signed**, which can be read by all mail clients.

This is a format string, see the \$smime\_decrypt\_command command for possible **printf(3)**-like sequences. NOTE: %c and %k will default to \$smime\_sign\_as if set, otherwise \$smime\_default\_key. (S/MIME only)

**smime\_sign\_digest\_alg**

Type: string

Default: "sha256"

This sets the algorithm that should be used for the signature message digest. Valid choices are "md5", "sha1", "sha224", "sha256", "sha384", "sha512". (S/MIME only)

**smime\_sign\_opaque\_command**

Type: string

Default: ""

This command is used to create S/MIME signatures of type **application/x-pkcs7-signature**, which can only be handled by mail clients supporting the S/MIME extension.

This is a format string, see the \$smime\_decrypt\_command command for possible **printf(3)**-like sequences. (S/MIME only)

**smime\_timeout**

Type: number (long)

Default: 300

The number of seconds after which a cached passphrase will expire if not used. (S/MIME only)

**smime\_verify\_command**

Type: string

Default: ""

This command is used to verify S/MIME signatures of type **multipart/signed**.

This is a format string, see the \$smime\_decrypt\_command command for possible **printf(3)**-like sequences. (S/MIME only)

**smime\_verify\_opaque\_command**

Type: string

Default: ""

This command is used to verify S/MIME signatures of type **application/x-pkcs7-mime**.

This is a format string, see the \$smime\_decrypt\_command command for possible **printf(3)**-like sequences. (S/MIME only)

**smtp\_authenticators**

Type: string

Default: ""

This is a colon-delimited list of authentication methods mutt may attempt to use to log in to an SMTP server, in the order mutt should try them. Authentication methods are any SASL mechanism, e.g. “digest-md5”, “gssapi” or “cram-md5”. This option is case-insensitive. If it is “unset” (the default) mutt will try all available methods, in order from most-secure to least-secure.

Example:

```
set smtp_authenticators="digest-md5:cram-md5"
```

### **smtp\_oauth\_refresh\_command**

Type: string

Default: “”

The command to run to generate an OAUTH refresh token for authorizing your connection to your SMTP server. This command will be run on every connection attempt that uses the OAUTH-BEARER authentication mechanism. See “oauth” for details.

### **smtp\_pass**

Type: string

Default: “”

Specifies the password for your SMTP account. If *unset*, Mutt will prompt you for your password when you first send mail via SMTP. See \$smtp\_url to configure mutt to send mail via SMTP.

**Warning:** you should only use this option when you are on a fairly secure machine, because the superuser can read your muttrc even if you are the only one who can read the file.

### **smtp\_url**

Type: string

Default: “”

Defines the SMTP smarthost where sent messages should relayed for delivery. This should take the form of an SMTP URL, e.g.:

```
smtp[s]://[user[:pass]@]host[:port]
```

where “[...]” denotes an optional part. Setting this variable overrides the value of the \$sendmail variable.

Also see \$write\_bcc.



**sort**

Type: sort order

Default: date

Specifies how to sort messages in the “index” menu. Valid values are:

- date or date–sent
- date–received
- from
- mailbox–order (unsorted)
- score
- size
- spam
- subject
- threads
- to

You may optionally use the “reverse–” prefix to specify reverse sorting order (example: “**set sort=reverse–date–sent**”).

**sort\_alias**

Type: sort order

Default: alias

Specifies how the entries in the “alias” menu are sorted. The following are legal values:

- address (sort alphabetically by email address)
- alias (sort alphabetically by alias name)
- unsorted (leave in order specified in .muttrc)

**sort\_aux**

Type: sort order

Default: date

This provides a secondary sort for messages in the “index” menu, used when the \$sort value is equal for two messages.

When sorting by threads, this variable controls how threads are sorted in relation to other threads, and how the branches of the thread trees are sorted. This can be set to any value that \$sort can, except “threads” (in that case, mutt will just use “date–sent”). You can also specify the “last–” prefix in addition to the “reverse–” prefix, but “last–” must come after “reverse–”. The “last–” prefix causes messages to be sorted against its siblings by which has the last descendant, using the rest of \$sort\_aux as an ordering. For instance,

```
set sort_aux=last-date-received
```

would mean that if a new message is received in a thread, that thread becomes the last one displayed (or the first, if you have “**set sort=reverse–threads**”).

Note: For reversed–threads \$sort order, \$sort\_aux is reversed again (which is not the right thing to do, but kept to not break any existing configuration setting).

**sort\_browser**

Type: sort order

Default: alpha

Specifies how to sort entries in the file browser. By default, the entries are sorted alphabetically.

Valid values:

- alpha (alphabetically)
- count
- date
- size
- unread
- unsorted

You may optionally use the “reverse–” prefix to specify reverse sorting order (example: “**set sort\_browser=reverse–date**”).

**sort\_browser\_mailboxes**

Type: sort order

Default: unsorted

Specifies how to sort entries in the mailbox browser. By default, the entries are unsorted, displayed in the same order as listed in the “mailboxes” command. Valid values:

- alpha (alphabetically)
- count
- date
- size
- unread
- unsorted

You may optionally use the “reverse–” prefix to specify reverse sorting order (example: “**set sort\_browser\_mailboxes=reverse–alpha**”).

**sort\_re**

Type: boolean

Default: yes

This variable is only useful when sorting by threads with `$strict_threads unset`. In that case, it changes the heuristic mutt uses to thread messages by subject. With `$sort_re set`, mutt will only attach a message as the child of another message by subject if the subject of the child message starts with a substring matching the setting of `$reply_regexp`. With `$sort_re unset`, mutt will attach the message whether or not this is the case, as long as the non-`$reply_regexp` parts of both messages are identical.

**spam\_separator**

Type: string

Default: “,”

This variable controls what happens when multiple spam headers are matched: if *unset*, each successive header will overwrite any previous matches value for the spam label. If *set*, each successive match will append to the previous, using this variable's value as a separator.

**spoolfile**

Type: path  
Default: ""

If your spool mailbox is in a non-default place where Mutt cannot find it, you can specify its location with this variable. Mutt will initially set this variable to the value of the environment variable **\$MAIL** or **\$MAILDIR** if either is defined.

**ssl\_ca\_certificates\_file**

Type: path  
Default: ""

This variable specifies a file containing trusted CA certificates. Any server certificate that is signed with one of these CA certificates is also automatically accepted. (GnuTLS only)

Example:

```
set ssl_ca_certificates_file=/etc/ssl/certs/ca-certificates.crt
```

**ssl\_client\_cert**

Type: path  
Default: ""

The file containing a client certificate and its associated private key.

**ssl\_force\_tls**

Type: boolean  
Default: yes

If this variable is *set*, Mutt will require that all connections to remote servers be encrypted. Furthermore it will attempt to negotiate TLS even if the server does not advertise the capability, since it would otherwise have to abort the connection anyway. This option supersedes `$ssl_starttls`.

**ssl\_min\_dh\_prime\_bits**

Type: number  
Default: 0

This variable specifies the minimum acceptable prime size (in bits) for use in any Diffie–Hellman key exchange. A value of 0 will use the default from the GNUTLS library. (GnuTLS only)

#### **ssl\_starttls**

Type: quadoption

Default: yes

If *set* (the default), mutt will attempt to use **STARTTLS** on servers advertising the capability. When *unset*, mutt will not attempt to use **STARTTLS** regardless of the server’s capabilities.

**Note** that **STARTTLS** is subject to many kinds of attacks, including the ability of a machine-in-the-middle to suppress the advertising of support. Setting `$ssl_force_tls` is recommended if you rely on **STARTTLS**.

#### **ssl\_use\_sslv2**

Type: boolean

Default: no

If *set*, Mutt will use SSLv2 when communicating with servers that request it. **N.B. As of 2011, SSLv2 is considered insecure, and using is inadvisable. See <https://tools.ietf.org/html/rfc6176>.** (OpenSSL only)

#### **ssl\_use\_sslv3**

Type: boolean

Default: no

If *set*, Mutt will use SSLv3 when communicating with servers that request it. **N.B. As of 2015, SSLv3 is considered insecure, and using it is inadvisable. See <https://tools.ietf.org/html/rfc7525>.**

#### **ssl\_use\_tlsv1**

Type: boolean

Default: no

If *set*, Mutt will use TLSv1.0 when communicating with servers that request it. **N.B. As of 2015, TLSv1.0 is considered insecure, and using it is inadvisable. See <https://tools.ietf.org/html/rfc7525>.**

#### **ssl\_use\_tlsv1\_1**

Type: boolean

Default: no

If *set*, Mutt will use TLSv1.1 when communicating with servers that request it. **N.B. As of 2015, TLSv1.1 is considered insecure, and using it is inadvisable. See**

<https://tools.ietf.org/html/rfc7525> .

**ssl\_use\_tlsv1\_2**

Type: boolean

Default: yes

If *set* , Mutt will use TLSv1.2 when communicating with servers that request it.

**ssl\_use\_tlsv1\_3**

Type: boolean

Default: yes

If *set* , Mutt will use TLSv1.3 when communicating with servers that request it.

**ssl\_usesystemcerts**

Type: boolean

Default: yes

If set to *yes*, mutt will use CA certificates in the system-wide certificate store when checking if a server certificate is signed by a trusted CA. (OpenSSL only)

**ssl\_verify\_dates**

Type: boolean

Default: yes

If *set* (the default), mutt will not automatically accept a server certificate that is either not yet valid or already expired. You should only unset this for particular known hosts, using the **<account-hook>** function.

**ssl\_verify\_host**

Type: boolean

Default: yes

If *set* (the default), mutt will not automatically accept a server certificate whose host name does not match the host used in your folder URL. You should only unset this for particular known hosts, using the **<account-hook>** function.

**ssl\_verify\_host\_override**

Type: string

Default: ""

Defines an alternate host name to verify the server certificate against. This should not be set unless you are sure what you are doing, but it might be useful for connection to a .onion host without a properly configured host name in the certificate. See `$ssl_verify_host`.

### **ssl\_verify\_partial\_chains**

Type: boolean

Default: no

This option should not be changed from the default unless you understand what you are doing.

Setting this variable to *yes* will permit verifying partial certification chains, i. e. a certificate chain where not the root, but an intermediate certificate CA, or the host certificate, are marked trusted (in `$certificate_file`), without marking the root signing CA as trusted.

(OpenSSL 1.0.2b and newer only).

### **ssl\_ciphers**

Type: string

Default: ""

Contains a colon-separated list of ciphers to use when using SSL. For OpenSSL, see `ciphers(1)` for the syntax of the string.

For GnuTLS, this option will be used in place of "NORMAL" at the start of the priority string. See `gnutls_priority_init(3)` for the syntax and more details. (Note: GnuTLS version 2.1.7 or higher is required.)

### **status\_chars**

Type: string

Default: "-\*%A"

Controls the characters used by the "%r" indicator in `$status_format`. The first character is used when the mailbox is unchanged. The second is used when the mailbox has been changed, and it needs to be resynchronized. The third is used if the mailbox is in read-only mode, or if the mailbox will not be written when exiting that mailbox (You can toggle whether to write changes to a mailbox with the `<toggle-write>` operation, bound by default to "%"). The fourth is used to indicate that the current folder has been opened in attach-message mode (Certain operations like composing a new mail, replying, forwarding, etc. are not permitted in this mode).

### **status\_format**

Type: string (localized)

Default: "-%r-Mutt: %f [Msgs:%?M?%M/?%m%?n? New:%n?%?o? Old:%o?%?d? Del:%d?%?F? Flag:%F?%

Controls the format of the status line displayed in the "index" menu. This string is similar to `$index_format`, but has its own set of **printf(3)**-like sequences:

%b	number of mailboxes with new mail *
%B	number of backgrounded editing sessions *
%d	number of deleted messages *
%f	the full pathname of the current mailbox
%F	number of flagged messages *
%h	local hostname
%l	size (in bytes) of the current mailbox (see formatstrings-size) *
%L	size (in bytes) of the messages shown (i.e., which match the current limit) (see formatstrings-size) *
%m	the number of messages in the mailbox *
%M	the number of messages shown (i.e., which match the current limit) *
%n	number of new messages in the mailbox *
%o	number of old unread messages *
%p	number of postponed messages *
%P	percentage of the way through the index
%r	modified/read-only/won't-write/attach-message indicator, according to \$status_chars
%R	number of read messages *
%s	current sorting mode (\$sort)
%S	current aux sorting method (\$sort_aux)
%t	number of tagged messages *
%u	number of unread messages *
%v	Mutt version string
%V	currently active limit pattern, if any *
%>X	right justify the rest of the string and pad with "X"
% X	pad to the end of the line with "X"
/*X	soft-fill with character "X" as pad

For an explanation of "soft-fill", see the \$index\_format documentation.

\* = can be optionally printed if nonzero

Some of the above sequences can be used to optionally print a string if their value is nonzero. For example, you may only want to see the number of flagged messages if such messages exist, since zero is not particularly meaningful. To optionally print a string based upon one of the above sequences, the following construct is used:

**%?<sequence\_char>?<optional\_string>?**

where *sequence\_char* is a character from the table above, and *optional\_string* is the string you would like printed if *sequence\_char* is nonzero. *optional\_string* **may** contain other sequences as well as normal text, but you may **not** nest optional strings.

Here is an example illustrating how to optionally print the number of new messages in a mailbox:

**%?n?%n new messages.?**

You can also switch between two strings using the following construct:

**%?<sequence\_char>?<if\_string>&<else\_string>?**

If the value of *sequence\_char* is non-zero, *if\_string* will be expanded, otherwise *else\_string* will be expanded.

You can force the result of any **printf(3)**-like sequence to be lowercase by prefixing the sequence character with an underscore (“\_”) sign. For example, if you want to display the local hostname in lowercase, you would use: “%\_h”.

If you prefix the sequence character with a colon (“:”) character, mutt will replace any dots in the expansion by underscores. This might be helpful with IMAP folders that don’t like dots in folder names.

#### **status\_on\_top**

Type: boolean  
Default: no

Setting this variable causes the “status bar” to be displayed on the first line of the screen rather than near the bottom. If \$help is *set*, too it’ll be placed at the bottom.

#### **strict\_threads**

Type: boolean  
Default: no

If *set*, threading will only make use of the “In-Reply-To” and “References:” fields when you \$sort by message threads. By default, messages with the same subject are grouped together in “pseudo threads.” This may not always be desirable, such as in a personal mailbox where you might have several unrelated messages with the subjects like “hi” which will get grouped together. See also \$sort\_re for a less drastic way of controlling this behavior.

#### **suspend**

Type: boolean  
Default: yes

When *unset*, mutt won’t stop when the user presses the terminal’s *susp* key, usually “Z”. This is useful if you run mutt inside an xterm using a command like “**xterm -e mutt**”.

#### **text\_flowed**

Type: boolean  
Default: no

When *set*, mutt will generate “format=flowed” bodies with a content type of “**text/plain; format=flowed**”. This format is easier to handle for some mailing software, and generally just looks like ordinary text. To actually make use of this format’s features, you’ll need support in your editor.

The option only controls newly composed messages. Postponed messages, resent messages, and draft messages (via -H on the command line) will use the content-type of the source message.

Note that \$indent\_string is ignored when this option is *set*.



**thorough\_search**

Type: boolean  
Default: yes

Affects the **~b**, **~B**, and **~h** search operations described in section “patterns”. If *set*, the headers and body/attachments of messages to be searched are decoded before searching. If *unset*, messages are searched as they appear in the folder.

Users searching attachments or for non-ASCII characters should *set* this value because decoding also includes MIME parsing/decoding and possible character set conversions. Otherwise mutt will attempt to match against the raw message received (for example quoted-printable encoded or with encoded headers) which may lead to incorrect search results.

**thread\_received**

Type: boolean  
Default: no

When *set*, mutt uses the date received rather than the date sent to thread messages by subject.

**tilde**

Type: boolean  
Default: no

When *set*, the internal-pager will pad blank lines to the bottom of the screen with a tilde (“~”).

**time\_inc**

Type: number  
Default: 0

Along with `$read_inc`, `$write_inc`, and `$net_inc`, this variable controls the frequency with which progress updates are displayed. It suppresses updates less than `$time_inc` milliseconds apart. This can improve throughput on systems with slow terminals, or when running mutt on a remote system.

Also see the “tuning” section of the manual for performance considerations.

**timeout**

Type: number  
Default: 600

When Mutt is waiting for user input either idling in menus or in an interactive prompt, Mutt would block until input is present. Depending on the context, this would prevent certain operations from working, like checking for new mail or keeping an IMAP connection alive.

This variable controls how many seconds Mutt will at most wait until it aborts waiting for input, performs these operations and continues to wait for input.

A value of zero or less will cause Mutt to never time out.

**tmpdir**

Type: path

Default: ""

This variable allows you to specify where Mutt will place its temporary files needed for displaying and composing messages. If this variable is not set, the environment variable **\$TMPDIR** is used. If **\$TMPDIR** is not set then **"/var/tmp"** is used.

**to\_chars**

Type: string

Default: "+TCFL"

Controls the character used to indicate mail addressed to you. The first character is the one used when the mail is *not* addressed to your address. The second is used when you are the only recipient of the message. The third is when your address appears in the "To:" header field, but you are not the only recipient of the message. The fourth character is used when your address is specified in the "Cc:" header field, but you are not the only recipient. The fifth character is used to indicate mail that was sent by *you*. The sixth character is used to indicate when a mail was sent to a mailing-list you subscribe to.

**trash**

Type: path

Default: ""

If set, this variable specifies the path of the trash folder where the mails marked for deletion will be moved, instead of being irremediably purged.

NOTE: When you delete a message in the trash folder, it is really deleted, so that you have a way to clean the trash.

**ts\_icon\_format**

Type: string (localized)

Default: "M%?n?AIL&ail?"

Controls the format of the icon title, as long as **"\$ts\_enabled"** is set. This string is identical in formatting to the one used by **"\$status\_format"**.

**ts\_enabled**

Type: boolean

Default: no

Controls whether mutt tries to set the terminal status line and icon name. Most terminal emulators emulate the status line in the window title.

**ts\_status\_format**

Type: string (localized)

Default: “Mutt with %?m?%m messages&no messages?%?n? [%n NEW]?”

Controls the format of the terminal status line (or window title), provided that “\$ts\_enabled” has been set. This string is identical in formatting to the one used by “\$status\_format”.

**tunnel**

Type: string

Default: “”

Setting this variable will cause mutt to open a pipe to a command instead of a raw socket. You may be able to use this to set up preauthenticated connections to your IMAP/POP3/SMTP server. Example:

```
set tunnel="ssh -q mailhost.net /usr/local/libexec/imapd"
```

Note: For this example to work you must be able to log in to the remote machine without having to enter a password.

When set, Mutt uses the tunnel for all remote connections. Please see “account-hook” in the manual for how to use different tunnel commands per connection.

**tunnel\_is\_secure**

Type: boolean

Default: yes

When *set*, Mutt will assume the \$tunnel connection does not need STARTTLS to be enabled. It will also allow IMAP PREAUTH server responses inside a tunnel to proceed. This is appropriate if \$tunnel uses ssh or directly invokes the server locally.

When *unset*, Mutt will negotiate STARTTLS according to the ssl\_starttls and ssl\_force\_tls variables. If ssl\_force\_tls is set, Mutt will abort connecting if an IMAP server responds with PREAUTH. This setting is appropriate if \$tunnel does not provide security and could be tampered with by attackers.

**uncollapse\_jump**

Type: boolean

Default: no

When *set*, Mutt will jump to the next unread message, if any, when the current thread is *uncollapsed*.

**uncollapse\_new**

Type: boolean

Default: yes

When *set*, Mutt will automatically uncollapse any collapsed thread that receives a new message. When *unset*, collapsed threads will remain collapsed. the presence of the new message will still affect index sorting, though.

**use\_8bitmime**

Type: boolean

Default: no

**Warning:** do not set this variable unless you are using a version of sendmail which supports the **-B8BITMIME** flag (such as sendmail 8.8.x) or you may not be able to send mail.

When *set*, Mutt will invoke \$sendmail with the **-B8BITMIME** flag when sending 8-bit messages to enable ESMTP negotiation.

**use\_domain**

Type: boolean

Default: yes

When *set*, Mutt will qualify all local addresses (ones without the “@host” portion) with the value of \$hostname. If *unset*, no addresses will be qualified.

**use\_envelope\_from**

Type: boolean

Default: no

When *set*, mutt will set the *envelope* sender of the message. If \$envelope\_from\_address is *set*, it will be used as the sender address. If *unset*, mutt will attempt to derive the sender from the “From:” header.

Note that this information is passed to sendmail command using the **-f** command line switch. Therefore setting this option is not useful if the \$sendmail variable already contains **-f** or if the executable pointed to by \$sendmail doesn’t support the **-f** switch.

**use\_from**

Type: boolean

Default: yes

When *set*, Mutt will generate the “From:” header field when sending messages. If *unset*, no “From:” header field will be generated unless the user explicitly sets one using the “my\_hdr” command.

**use\_ipv6**

Type: boolean

Default: yes

When *set*, Mutt will look for IPv6 addresses of hosts it tries to contact. If this option is *unset*, Mutt will restrict itself to IPv4 addresses. Normally, the default should work.

**user\_agent**

Type: boolean

Default: no

When *set*, mutt will add a “User-Agent:” header to outgoing messages, indicating which version of mutt was used for composing them.

**visual**

Type: path

Default: “”

Specifies the visual editor to invoke when the “~v” command is given in the built-in editor.

**wait\_key**

Type: boolean

Default: yes

Controls whether Mutt will ask you to press a key after an external command has been invoked by these functions: **<shell-escape>**, **<pipe-message>**, **<pipe-entry>**, **<print-message>**, and **<print-entry>** commands.

It is also used when viewing attachments with “auto\_view”, provided that the corresponding mailcap entry has a *needsterminal* flag, and the external program is interactive.

When *set*, Mutt will always ask for a key. When *unset*, Mutt will wait for a key only if the external command returned a non-zero status.

**weed**

Type: boolean

Default: yes

When *set*, mutt will weed headers when displaying, forwarding, or replying to messages.

Also see `$copy_decode_weed`, `$pipe_decode_weed`, `$print_decode_weed`.

**wrap**

Type: number

Default: 0

When set to a positive value, mutt will wrap text at `$wrap` characters. When set to a negative value, mutt will wrap text so that there are `$wrap` characters of empty space on the right side of the terminal. Setting it to zero makes mutt wrap at the terminal width.

Also see `$reflow_wrap`.

**wrap\_headers**

Type: number

Default: 78

This option specifies the number of characters to use for wrapping an outgoing message's headers. Allowed values are between 78 and 998 inclusive.

**Note:** This option usually shouldn't be changed. RFC5233 recommends a line length of 78 (the default), so **please only change this setting when you know what you're doing**.

**wrap\_search**

Type: boolean

Default: yes

Controls whether searches wrap around the end.

When *set*, searches will wrap around the first (or last) item. When *unset*, incremental searches will not wrap.

**wrapmargin**

Type: number

Default: 0

(DEPRECATED) Equivalent to setting `$wrap` with a negative value.

**write\_bcc**

Type: boolean

Default: no

Controls whether mutt writes out the "Bcc:" header when preparing messages to be sent. Some MTAs, such as Exim and Courier, do not strip the "Bcc:" header; so it is advisable to leave this *unset* unless you have a particular need for the header to be in the sent message.

If mutt is set to deliver directly via SMTP (see `$smtp_url`), this option does nothing: mutt will never write out the “Bcc:” header in this case.

Note this option only affects the sending of messages. Fcc’ed copies of a message will always contain the “Bcc:” header if one exists.

**write\_inc**

Type: number

Default: 10

When writing a mailbox, a message will be printed every `$write_inc` messages to indicate progress. If set to 0, only a single message will be displayed before writing a mailbox.

Also see the `$read_inc`, `$net_inc` and `$time_inc` variables and the “tuning” section of the manual for performance considerations.

**SEE ALSO**

**iconv(1), iconv(3), mailcap(5), maildir(5), mbox(5), mutt(1), printf(3), regex(7), strftime(3)**

The Mutt Manual

The Mutt home page: <http://www.mutt.org/>

**AUTHOR**

Michael Elkins, and others. Use `<mutt-dev@mutt.org>` to contact the developers.