The MH-E Manual

Version 8.0.3
November, 2006

Bill Wohler

Table of Contents

Preface

This manual introduces another interface to the MH mail system that is accessible through the GNU Emacs editor, namely, *MH-E*. MH-E is easy to use. I don't assume that you know GNU Emacs or even MH at this point, since I didn't know either of them when I discovered MH-E. However, MH-E was the tip of the iceberg, and I discovered more and more niceties about GNU Emacs and MH. Now I'm fully hooked on both of them.

The MH-E package is distributed with GNU Emacs[1], so you shouldn't have to do anything special to use it. This manual covers MH-E version 8.0.3. To help you decide which version you have, see Chapter 2 [Getting Started], page 5.

If you don't already use GNU Emacs but want to learn more, you can read an online tutorial by starting GNU Emacs and typing `C-h t` (`help-with-tutorial`). (To learn about this notation, see Chapter 1 [Conventions], page 3.) If you want to take the plunge, consult the *GNU Emacs Manual*, from the Free Software Foundation.

If more information is needed, you can go to the Unix manual pages of the individual MH commands. When the name is not obvious, I'll guide you to a relevant MH manual page that describes the action more fully.

This manual is available in both Info and online formats. The Info version is distributed with Emacs and can be accessed with the `info` command ('`info mh-e`') or within Emacs (`C-h i m mh-e` ⟨RET⟩). The online version is available at SourceForge. Another great online resource is the book *MH & nmh: Email for Users & Programmers* (also known as *the MH book*).

I hope you enjoy this manual! If you have any comments, or suggestions for this document, please let me know.

Bill Wohler <*wohler at newt.com*>
8 February 1995
24 February 2006

---

[1] Version 8.0.3 of MH-E will appear in GNU Emacs 22.1. It is supported in GNU Emacs 21, as well as XEmacs 21 (except for versions 21.5.9-21.5.16). It is compatible with MH versions 6.8.4 and higher, all versions of nmh, and GNU mailutils 1.0 and higher.

1 GNU Emacs Terms and Conventions

If you're an experienced Emacs user, you can skip the following conventions and definition of terms and go directly to the next section (see Chapter 2 [Getting Started], page 5).

In general, *functions* in this text refer to Emacs Lisp functions that one would call from within Emacs Lisp programs (for example, `(mh-inc-folder)`). On the other hand, *commands* are those things that are run by the user, such as `i` or `M-x mh-inc-folder`. Programs outside of Emacs are specifically called MH commands, shell commands, or Unix commands.

The conventions for key names are as follows:

`C-x`   Hold down the ⟨CTRL⟩ (Control) key and press the `x` key.

`M-x`   Hold down the ⟨META⟩ or ⟨ALT⟩ key and press the `x` key.

                 Since some keyboards don't have a ⟨META⟩ key, you can generate `M-x`, for example, by pressing ⟨ESC⟩ (Escape), *releasing it*, and then pressing the `x` key.

⟨RET⟩   Press the ⟨RETURN⟩ or ⟨ENTER⟩ key. This is normally used to complete a command.

⟨SPC⟩   Press the space bar.

⟨TAB⟩   Press the ⟨TAB⟩ key.

⟨DEL⟩   Press the ⟨DELETE⟩ key.

⟨BS⟩   Press the ⟨BACKSPACE⟩ key[1].

A *prefix argument* allows you to pass an argument to any Emacs function. To pass an argument, type `C-u` before the Emacs command or keystroke. Numeric arguments can be passed as well. For example, to insert five f's, use `C-u 5 f`. There is a default of four when using `C-u`, and you can use multiple prefix arguments to provide arguments of powers of four. To continue our example, you could insert four f's with `C-u f`, 16 f's with `C-u C-u f`, 64 f's with `C-u C-u C-u f`, and so on. Numeric and valueless negative arguments can also be inserted with the ⟨META⟩ key. Examples include `M-5` to specify an argument of 5, or `M--` which specifies a negative argument with no particular value.

NOTE

The prefix `C-u` or `M-` is not necessary in MH-E's MH-Folder mode (see Section 3.2 [Reading Mail Tour], page 8). In this mode, simply enter the numerical argument before entering the command.

Emacs uses *variables* to hold values. These can be changed via calls to the function `setq` in '`~/.emacs`'.

Variables in MH-E that are normally modified by the user are called *options* and are modified through the customize functions (such as `M-x customize-option` or `M-x customize-group`). See section "Easy Customization" in *The GNU Emacs Manual*, in *The GNU Emacs Manual*. See Section 4.1 [Options], page 11.

---

[1] If you are using Version 20 or earlier of Emacs, you will need to use the ⟨DEL⟩ key.

You can specify various styles for displaying text using *faces*. MH-E provides a set of faces that you can use to personalize the look of your MH-E buffers. Use the command `M-x customize-face` to do this. See section "Face Customization" in *The GNU Emacs Manual*, in *The GNU Emacs Manual*.

Commands often offer *hooks* which enable you to extend or modify the way a command works. section "Hooks" in *The GNU Emacs Manual*, in *The GNU Emacs Manual* for a description about *normal hooks* and *abnormal hooks*. MH-E uses normal hooks in nearly all cases, so you can assume that we are talking about normal hooks unless we explicitly mention that a hook is abnormal. We also follow the conventions described in that section: the name of the abnormal hooks end in `-hooks` and all the rest of the MH-E hooks end in `-hook`. You can add hooks with either `customize-option` or `add-hook`.

There are several other terms that are used in Emacs that you should know. The *point* is where the cursor currently is. You can save your current place in the file by setting a *mark*. This operation is useful in several ways. The mark can be later used when defining a *region*, which is the text between the point and mark. Many commands operate on regions, such as those for deleting text or filling paragraphs. A mark can be set with `C-@` (or `C-`⟨SPC⟩).

The *minibuffer* is the bottom line of the Emacs window, where all prompting and multiple-character input is directed. You can use *completion* to enter values such as folders. Completion means that Emacs fills in text for you when you type ⟨SPC⟩ or ⟨TAB⟩. A second ⟨SPC⟩ or ⟨TAB⟩ will list all possibilities at that point. See section "Completion" in *The GNU Emacs Manual*. Note that ⟨SPC⟩ cannot be used for completing filenames and folders.

The minibuffer is also where you enter Emacs function names after typing `M-x`. For example, in the preface, I mentioned that you could obtain help with `C-h t` (`help-with-tutorial`). What this means is that you can get a tutorial by typing either `C-h t` or `M-x help-with-tutorial`. In the latter case, you are prompted for '`help-with-tutorial`' in the minibuffer after typing `M-x`.

The '`~`' notation in filenames represents your home directory. This notation is used by many shells including `bash`, `tcsh`, and `csh`. It is analogous to the environment variable '`$HOME`'. For example, '`~/.emacs`' can be written '`$HOME/.emacs`' or using the absolute path as in '`/home/wohler/.emacs`' instead.

In case of trouble: Emacs can be interrupted at any time with `C-g`. For example, if you've started a command that requests that you enter something in the minibuffer, but then you change your mind, type `C-g` and you'll be back where you started. If you want to exit Emacs entirely, use `C-x C-c`.

2 Getting Started

Because there are many old versions of MH-E out there, it is important to know which version you have. I'll be talking about Version 8 which is pretty close to Version 6 and Version 7. It differs from Version 4 and Version 5 and is vastly different from Version 3. See Appendix D [History], page 101.

To determine which version of MH-E that you have, enter `M-x mh-version` (RET). Hopefully it says that you're running Version 8.0.3 which is the latest version as of this printing.

If your version is much older than this, please consider upgrading. You can have your system administrator upgrade the system-wide version, or you can install your own personal version. It's really quite easy. See Section C.4 [Getting MH-E], page 99, for instructions for getting and installing MH-E.

If the `mh-version` command displays 'No MH variant detected'[1], then you need to install MH or tell MH-E where to find MH.

If you don't have MH on your system already, you must install a variant of MH. The Debian mh-e package does this for you automatically (see Section C.4 [Getting MH-E], page 99). Most people use nmh, but you may be interested in trying out GNU mailutils, which supports IMAP. Your GNU/Linux distribution probably has packages for both of these.

If you've never run MH before, you need to run `install-mh` from the shell before you continue. This sets up your personal MH environment[2]. If you don't, you'll be greeted with the error message: 'Install MH and run install-mh before running MH-E'. This is all you need to know about MH to use MH-E, but the more you know about MH, the more you can leverage its power. See the MH book to learn more about MH.

Your MH environment includes your *MH profile* which is found in the file '~/.mh_profile'. This file contains a number of *MH profile components*. For example, the 'Path:' MH profile component contains the path to your mail directory, which is '~/Mail' by default.

There are several options MH-E uses to interact with your MH installation. The option `mh-variant` specifies the variant used by MH-E (see Section 4.1 [Options], page 11). The default setting of this option is 'Auto-detect' which means that MH-E will automatically choose the first of nmh, MH, or GNU mailutils that it finds in the directories listed in `mh-path` (which you can customize), `mh-sys-path`, and `exec-path`. If MH-E can't find MH at all, you may have to customize `mh-path` and add the directory in which the command `mhparam` is located. If, on the other hand, you have both nmh and mailutils installed (for example) and `mh-variant-in-use` was initialized to nmh but you want to use mailutils, then you can set `mh-variant` to 'mailutils'.

When `mh-variant` is changed, MH-E resets `mh-progs`, `mh-lib`, `mh-lib-progs`, `mh-flists-present-flag`, and `mh-variant-in-use` accordingly.

---

[1] In very old versions of MH-E, you may get the error message, 'Cannot find the commands 'inc' and 'mhl' and the file 'components'' if MH-E can't find MH. In this case, you need to update MH-E, and you may need to install MH too. However, newer versions of MH-E are better at finding MH if it is on your system.

[2] See the section Setting Up MH in the MH book.

NOTE

Prior to version 8, it was often necessary to set some of these variables in
'`~/.emacs`'; now it is no longer necessary and can actually cause problems.

In addition to setting variables that point to MH itself, MH-E also sets a handful of
variables that point to where you keep your mail. During initialization, the function `mh-find-path` sets `mh-user-path` from your '`Path:`' MH profile component (but defaults to
'`Mail`' if one isn't present), `mh-draft-folder` from '`Draft-Folder:`', `mh-unseen-seq` from
'`Unseen-Sequence:`', `mh-previous-seq` from '`Previous-Sequence:`', and `mh-inbox` from
'`Inbox:`' (defaults to '`+inbox`'). The hook `mh-find-path-hook` is run after these variables
have been set. This hook can be used the change the value of these variables if you need to
run with different values between MH and MH-E.

3 Tour Through MH-E

This chapter introduces some of the terms you'll need to know and then takes you on a tour of MH-E[1]. When you're done, you'll be able to send, read, and file mail, which is all that a lot of people ever do. But if you're the curious or adventurous type, read the rest of the manual to be able to use all the features of MH-E. I suggest you read this chapter first to get the big picture, and then you can read the manual as you wish.

3.1 Sending Mail

Let's start our tour by sending ourselves a message which we can later read and process. Enter `M-x mh-smail` to invoke the MH-E program to send messages. Your message appears in an Emacs buffer whose mode[2] is MH-Letter.

Enter your login name in the 'To:' header field. Press the ⟨TAB⟩ twice to move the cursor past the 'Cc:' field, since no carbon copies are to be sent, and on to the 'Subject:' field. Enter *Test* or anything else that comes to mind.

Press ⟨TAB⟩ again to move the cursor to the body of the message. Enter some text, using normal Emacs commands. You should now have something like this[3]:

```
--:--  *scratch*   All L1     (Lisp Interaction)-----------------------
To: wohler
cc:
Subject: Test
X-Mailer: MH-E 8.0; nmh 1.1; GNU Emacs 22.1
--------
This is a test message to get the wheels churning...#


--:** {draft}   All L5     (MH-Letter)--------------------------------
Type C-c C-c to send message, C-C ? for help
```

MH-E message composition window

Note the line of dashes that separates the header and the body of the message. It is essential that these dashes (or a blank line) are present or the body of your message will be considered to be part of the header.

There are several commands specific to MH-Letter mode[4], but at this time we'll only use `C-c C-c` to send your message. Type `C-c C-c` now. That's all there is to it!

---

[1] The keys mentioned in these chapters refer to the default key bindings. If you've changed the bindings, refer to the command summaries at the beginning of each chapter for a mapping between default key bindings and function names.

[2] A *mode* changes Emacs to make it easier to edit a particular type of text.

[3] If you're running Emacs under the X Window System, then you would also see a menu bar and a tool bar. I've left out the menu bar and tool bar in all of the example screens.

[4] You can get quick help for the commands used most often with `C-c ?` or more complete help with the `C-h m` (describe-mode) command.

3.2 Receiving Mail

To read the mail you've just sent yourself, enter *M-x mh-rmail*. This incorporates the new mail and puts the output from `inc`[5] (called *scan lines* after the MH program `scan`[6] which prints a one-line summary of each message) into a buffer called '`+inbox`' whose major mode is MH-Folder.

NOTE

The *M-x mh-rmail* command will show you only new mail, not mail you have already read. If you were to run this tour again, you would use *F r* to pull all your messages into MH-E.

You should see the scan line for your message, and perhaps others. Use *n* or *p* to move the cursor to your test message and type ⟨RET⟩ to read your message. You should see something like:

```
  3 t08/24 root       received fax files on Wed Aug 24 11:00:13 PDT 1
# 4+t08/24 To:wohler  Test<<This is a test message to get the wheels

-:%%  {+inbox/select} 4 msgs (1-4)   Bot L4     (MH-Folder Show)---------
To: wohler
Subject: Test
X-Mailer: MH-E 8.0; nmh 1.1; GNU Emacs 22.1
Date: Fri, 17 Mar 2006 10:49:11 -0800
From: Bill Wohler <wohler@stop.mail-abuse.org>

This is a test message to get the wheels churning...




--:--  {show-+inbox} 4   All L1     (MH-Show)---------------------------
```

After incorporating new messages

If you typed a long message, you can view subsequent pages with ⟨SPC⟩ and previous pages with ⟨DEL⟩.

3.3 Processing Mail

The first thing we want to do is reply to the message that we sent ourselves. Ensure that the cursor is still on the same line as your test message and type *r*. You are prompted in the minibuffer with '`Reply to whom:`'. Here MH-E is asking whether you'd like to reply to the original sender only, to the sender and primary recipients, or to the sender and all recipients. You can press ⟨TAB⟩ to see these choices. If you simply press ⟨RET⟩, you'll reply only to the sender. Press ⟨RET⟩ now.

You'll find yourself in an Emacs buffer similar to that when you were sending the original message, like this:

---

[5] See the section Reading Mail: inc show next prev in the MH book.
[6] See the section Find and Specify with scan pick Ranges Sequences in the MH book.

```
To:
cc:
Subject: Re: Test
In-reply-to: <31054.1142621351@stop.mail-abuse.org>
References: <31054.1142621351@stop.mail-abuse.org>
Comments: In-reply-to Bill Wohler <wohler@stop.mail-abuse.org>
   message dated "Fri, 17 Mar 2006 10:49:11 -0800."
X-Mailer: MH-E 8.0; nmh 1.1; GNU Emacs 22.1
--------
#


--:-- {draft}  All L10     (MH-Letter)-------------------------------
To: wohler
Subject: Test
X-Mailer: MH-E 8.0; nmh 1.1; GNU Emacs 22.1
Date: Fri, 17 Mar 2006 10:49:11 -0800
From: Bill Wohler <wohler@stop.mail-abuse.org>

This is a test message to get the wheels churning...

--:-- {show-+inbox} 4   All L1     (MH-Show)--------------------------
Type C-c C-c to send message, C-c ? for help
```

Composition window during reply

By default, MH will not add you to the address list of your replies, so if you find that the 'To:' header field is missing, don't worry. In this case, type *C-c C-f C-t* to create and go to the 'To:' field, where you can type your login name again. You can move around with the arrow keys or with *C-p* (previous-line), *C-n* (next-line), *C-b* (backward-char), and *C-f* (forward-char) and can delete the previous character with (BS). When you're finished editing your message, send it with *C-c C-c* as before.

You'll often want to save messages that were sent to you in an organized fashion. This is done with *folders*. You can use folders to keep messages from your friends, or messages related to a particular topic. With your cursor in the MH-Folder buffer and positioned on the message you sent to yourself, type *o* to output (refile in MH parlance) that message to a folder. Enter *test* at the 'Destination folder:' prompt and type *y* (or (SPC)) when MH-E asks to create the folder '+test'. Note that a '^' (caret) appears next to the message number, which means that the message has been marked for refiling but has not yet been refiled. We'll talk about how the refile is actually carried out in a moment.

Your previous reply is now waiting in the system mailbox. You incorporate this mail into your MH-Folder buffer named '+inbox' with the *i* command. Do this now. After the mail is incorporated, use *n* or *p* to move the cursor to the new message, and read it with (RET). Let's delete this message by typing *d*. Note that a 'D' appears next to the message number. This means that the message is marked for deletion but is not yet deleted. To perform the deletion (and the refile we did previously), use the *x* command.

If you want to send another message you can use *m* instead of *M-x mh-smail*. So go ahead, send some mail to your friends!

You can get a quick reminder about these commands by typing *?*. This lists several *prefix characters*. To list the commands available via the prefix characters, type the prefix character followed by a *?*, for example, *F ?*. More complete help is available with the *C-h m* (describe-mode) command.

3.4 Leaving MH-E

You may now wish to exit `emacs` entirely. Use `C-x C-c` to exit `emacs`. If you exited without running `x` in the '`+inbox`' buffer, Emacs will offer to save it for you. Type `y` or (SPC) to save '`+inbox`' changes, which means to perform any refiles and deletes that you did there.

   If you don't want to leave Emacs, you can type `q` to bury (hide) the MH-E folder or delete it entirely with `C-x k`. You can then later recall it with `C-x b` or `M-x mh-rmail`.

   On the other hand, if you no longer want to use MH and MH-E, you can take your mail with you. You can copy all of your mail into a single file, mbox-style, by using the MH command `packf`. For example, to create a file called '`msgbox`' with the messages in your '`+inbox`' folder, use '`packf +inbox`'. The `packf` command will append the messages to the file if it already exists, so you can use '`folders -recurse -fast`' in a script to copy all of your messages into a single file, or using the '`-file`' argument, a file for each folder.

3.5 More About MH-E

These are the basic commands to get you going, but there are plenty more. If you think that MH-E is for you, read the rest of the manual to find out how you can:

- Print your messages (see Section 6.6 [Printing], page 28).
- Edit messages and include your signature (see Chapter 9 [Editing Drafts], page 45).
- Forward messages (see Section 8.3 [Forwarding], page 42).
- Read digests (see Section 6.4 [Digests], page 27).
- Edit bounced messages (see Section 8.5 [Editing Again], page 43).
- Send multimedia messages (see Section 9.6 [Adding Attachments], page 52).
- Read HTML messages (see Section 6.3 [HTML], page 25).
- Use aliases and identities (see Chapter 10 [Aliases], page 57, see Chapter 11 [Identities], page 61).
- Create different views of your mail (see Chapter 16 [Threading], page 77, see Chapter 17 [Limits], page 79).
- Deal with junk mail (see Chapter 19 [Junk], page 83).
- Handle signed and encrypted messages (see Section 6.5 [Reading PGP], page 27, see Section 9.7 [Sending PGP], page 55).
- Process mail that was sent with `shar` or `uuencode` (see Section 6.7 [Files and Pipes], page 29).
- Use sequences conveniently (see Chapter 18 [Sequences], page 81).
- Use the speedbar, tool bar, and menu bar (see Chapter 12 [Speedbar], page 65, see Chapter 14 [Tool Bar], page 69, see Chapter 13 [Menu Bar], page 67).
- Show header fields in different fonts (see Chapter 6 [Reading Mail], page 17).
- Find previously refiled messages (see Chapter 15 [Searching], page 71).
- Place messages in a file (see Section 6.7 [Files and Pipes], page 29).

   Remember that you can also use MH commands when you're not running MH-E (and when you are!).

4 Using This Manual

This chapter begins the meat of the manual which goes into more detail about every MH-E command and option.

There are many commands, but don't get intimidated. There are command summaries at the beginning of each chapter. In case you have or would like to rebind the keys, the command summaries also list the associated Emacs Lisp function. Furthermore, even if you're stranded on a desert island with a laptop and are without your manuals, you can get a summary of all these commands with GNU Emacs online help: use `C-h m` (`describe-mode`) for a brief summary of commands, `?` (`mh-help`) for an even briefer summary[1] (`C-c ?` in MH-Letter mode), or `C-h i` to read this manual via Info. The online help is quite good; try running `C-h C-h`. This brings up a list of available help topics, one of which displays the documentation for a given key (like `C-h k C-n`). Another useful help feature is to view the manual section that describes a given key (such as `C-h K i`). In addition, review Chapter 1 [Conventions], page 3, if any of the GNU Emacs conventions are strange to you.

In addition to all of the commands, it is also possible to reconfigure MH-E to fit the needs of even the most demanding user. The following chapters also describe all of the options, show the defaults, and make recommendations for customization.

However, when customizing your mail environment, first try to change what you want in MH, and only change MH-E if changing MH is not possible. That way you will get the same behavior inside and outside GNU Emacs. Note that MH-E does not provide hooks for customizations that can be done in MH; this omission is intentional.

I hope I've included enough examples here to get you well on your way. If you want to explore Emacs Lisp further, a programming manual does exist,[2] and you can look at the code itself for examples. Look in the Emacs Lisp directory on your system (such as '`/usr/local/lib/emacs/lisp/mh-e`') and find all the '`mh-*.el`' files there. When calling MH-E and other Emacs Lisp functions directly from Emacs Lisp code, you'll need to know the correct arguments. Use the online help for this. For example, try `C-h f mh-execute-commands` ⟨RET⟩. If you write your own functions, please do not prefix your symbols (variables and functions) with '`mh-`'. This prefix is reserved for the MH-E package. To avoid conflicts with existing MH-E symbols, use a prefix like '`my-`' or your initials. (Unless, of course, your initials happen to be *mh*!)

4.1 Options

Many string or integer options are easy to modify using `M-x customize-option`. For example, to modify the option that controls printing, you would run `M-x customize-option` ⟨RET⟩ `mh-lpr-command-format` ⟨RET⟩. In the buffer that appears, modify the string to the right of the variable. For example, you may change the `lpr` command with '`nenscript -G`

---

[1] This help appears in a buffer called '`*MH-E Help*`' (see Chapter 20 [Miscellaneous], page 89).

[2] The *GNU Emacs Lisp Reference Manual* may be available online in the Info system by typing `C-h i m Emacs Lisp` ⟨RET⟩. It is also available online at
http://www.gnu.org/software/emacs/elisp-manual/html_node/. You can also order a printed manual, which has the desirable side-effect of helping to support the Free Software Foundation which made all this great software available. You can find an order form by running `C-h C-d`, or you can request an order form from *gnu at gnu.org*.

`-r -2 -i'%s'`'. Then use the 'State' combo box and select 'Save for Future Sessions'. To read more about `mh-lpr-command-format`, see Section 6.6 [Printing], page 28.

Options can also hold boolean values. In Emacs Lisp, the boolean values are `nil`, which means false, and `t`, which means true. The `customize-option` function makes it easy to change boolean values; simply click on the toggle button in the customize buffer to switch between 'on' (`t`) and 'off' (`nil`). For example, try setting `mh-bury-show-buffer-flag` to 'off' to keep the MH-Show buffer at the top of the buffer stack. Use the 'State' combo box and choose 'Set for Current Session' to see how the option affects the show buffer. Then choose the 'Erase Customization' menu item to reset the option to the default, which places the MH-Show buffer at the bottom of the buffer stack.

The text usually says to turn on an option by setting it to a *non*-`nil` value, because sometimes values other than 'on' are meaningful. An example of this is the variable `mh-mhl-format-file` (see Section 6.1 [Viewing], page 21). Other options, such as hooks, involve a little more Emacs Lisp programming expertise.

You can browse all of the MH-E options with the `customize-group` function. Try entering `M-x customize-group` ⟨RET⟩ `mh` ⟨RET⟩ to view the top-level options as well as buttons for all of the MH-E customization groups. Another way to view the MH-E customization group is to use `M-x mh-customize` ⟨RET⟩.

4.2 Ranges

Many commands that operate on individual messages, such as `mh-forward` or `mh-refile-msg` take a `RANGE` argument. This argument can be used in several ways.

If you provide the prefix argument `C-u` to these commands, then you will be prompted for the message range. This can be any valid MH range which can include messages, sequences (see Chapter 18 [Sequences], page 81), and the abbreviations (described in the `mh`(1) man page):

'`<num1>-<num2>`'
>Indicates all messages in the range <num1> to <num2>, inclusive. The range must be nonempty.

'`<num>:N`'
'`<num>:+N`'
'`<num>:-N`'
>Up to N messages beginning with (or ending with) message num. Num may be any of the predefined symbols: first, prev, cur, next or last.

'`first:N`'
'`prev:N`'
'`next:N`'
'`last:N`' The first, previous, next or last messages, if they exist.

'`all`'   All of the messages.

For example, a range that shows all of these things is '1 2 3 5-10 last:5 unseen'.

If the option `transient-mark-mode` is turned on and you set a region in the MH-Folder buffer, then the MH-E command will perform the operation on all messages in that region.

The 'mh-range' customization group contains a single option which affects how ranges are interpreted.

mh-interpret-number-as-range-flag

>On means interpret a number as a range (default: 'on').

Since one of the most frequent ranges used is 'last:N', MH-E will interpret input such as '200' as 'last:200' if the mh-interpret-number-as-range-flag option is on (which is the default). If you need to scan just the message 200, then use the range '200:1' or '200-200'.

4.3 Folder Selection

When you choose a folder in MH-E via a command such as o (mh-refile-msg), completion is used to enter the folder (see section "Completion" in *The GNU Emacs Manual*). In addition, MH-E has several ways of choosing a suitable default so that the folder can often be selected with a single (RET) key.

The 'mh-folder-selection' customization group contains some options which are used to help with this.

mh-default-folder-for-message-function

>Function to select a default folder for refiling or 'Fcc:' (default: nil).

mh-default-folder-list

>List of addresses and folders (default: nil).

mh-default-folder-must-exist-flag

>On means guessed folder name must exist to be used (default: 'on').

mh-default-folder-prefix

>Prefix used for folder names generated from aliases (default: "").

You can set the option mh-default-folder-for-message-function to a function that provides a default folder for the message to be refiled. When this function is called, the current buffer contains the message being refiled and point is at the start of the message. This function should return the default folder as a string with a leading '+' sign. It can also return nil so that the last folder name is used as the default, or an empty string to suppress the default entirely.

Otherwise, the name of the destination folder is derived from the sender as follows:

1. The folder name associated with the first address found in the list mh-default-folder-list is used. Each element in this list contains a 'Check Recipient' item. If this item is turned on, then the address is checked against the recipient instead of the sender. This is useful for mailing lists.

2. An alias prefixed by mh-default-folder-prefix corresponding to the address is used. The prefix is used to prevent clutter in your mail directory. See Chapter 10 [Aliases], page 57.

If the derived folder does not exist, and mh-default-folder-must-exist-flag is t, then the last folder name used is suggested. This is useful if you get mail from various people for whom you have an alias, but file them all in the same project folder.

5 Incorporating Your Mail

This chapter talks about getting mail from your system mailbox into your MH '+inbox' folder. The following command accomplishes that and is found in the 'Folder' menu.

i   Incorporate new mail into a folder (mh-inc-folder).

The following options in the 'mh-inc' customization group are used.

mh-inc-prog
          Program to incorporate mail (default: "inc").

mh-inc-spool-list
          Alternate spool files (default: nil).

The following hook is available.

mh-inc-folder-hook
          Hook run by mh-inc-folder after incorporating mail into a folder (default:
          nil).

If at any time you receive new mail, incorporate the new mail into your '+inbox' buffer with i (mh-inc-folder). Note that i will display the '+inbox' buffer, even if there isn't any new mail. You can incorporate mail from any file into the current folder by specifying a prefix argument; you'll be prompted for the name of the file to use as well as the destination folder (for example, C-u i ~/mbox ⟨RET⟩ +tmp ⟨RET⟩).

Emacs can notify you when you have new mail by displaying 'Mail' in the mode line. To enable this behavior, and to have a clock in the mode line as well, add the following to '~/.emacs':

          (display-time)

The name of the program that incorporates new mail is stored in mh-inc-prog; it is "inc" by default. This program generates a one-line summary for each of the new messages. Unless it is an absolute pathname, the file is assumed to be in the mh-progs directory (see Chapter 2 [Getting Started], page 5). You may also link a file to inc that uses a different format (see 'mh-profile'(5), and sections Reading Mail: inc show next prev and MH Format Strings in the MH book). You'll then need to modify several variables appropriately (see Appendix A [Scan Line Formats], page 91).

You can use the mh-inc-spool-list variable to direct MH-E to retrieve mail from arbitrary spool files other than your system mailbox, file it in folders other than your '+inbox', and assign key bindings to incorporate this mail.

Suppose you are subscribed to the *mh-e-devel* mailing list and you use procmail to filter this mail into '~/mail/mh-e' with the following recipe in '.procmailrc':

```
PATH=$PATH:/usr/bin/mh
MAILDIR=$HOME/`mhparam Path`
:0:
* ^From mh-e-devel-admin@stop.mail-abuse.org
mh-e
```

In order to incorporate '~/mail/mh-e' into '+mh-e' with an I m (mh-inc-spool-mh-e) command, customize this option, and click on the 'INS' button. Enter a 'Spool File' of '~/mail/mh-e', a 'Folder' of 'mh-e', and a 'Key Binding' of 'm'.

You can use `xbuffy` to automate the incorporation of this mail using the Emacs 22 command `emacsclient` as follows:

```
box ~/mail/mh-e
    title mh-e
    origMode
    polltime 10
    headertime 0
    command emacsclient --eval '(mh-inc-spool-mh-e)'
```

In XEmacs, the command `gnuclient` is used in a similar fashion.

You can set the hook `mh-inc-folder-hook`, which is called after new mail is incorporated by the *i* (`mh-inc-folder`) command. A good use of this hook is to rescan the whole folder either after running *M-x mh-rmail* the first time or when you've changed the message numbers from outside of MH-E.

```
(defun my-mh-inc-folder-hook ()
  "Hook to rescan folder after incorporating mail."
  (if (buffer-modified-p)             ; if outstanding refiles and deletes,
      (mh-execute-commands))          ;    carry them out
  (mh-rescan-folder)                  ; synchronize with +inbox
  (mh-show))                          ; show the current message

(add-hook 'mh-inc-folder-hook 'my-mh-inc-folder-hook)

Rescan folder after incorporating new mail via mh-inc-folder-hook
```

6 Reading Your Mail

The MH-E entry point for reading mail is *M-x mh-rmail*. This command incorporates your mail and creates a buffer called '+inbox' in MH-Folder mode. The command *M-x mh-rmail* shows you only new mail, not mail you have already read[1].

There are some commands that need to read mail, such as *Mouse-2* over the 'Mail' button that `display-time` adds to the mode line. You can configure Emacs to have these commands use MH-E by setting the option `read-mail-command` to 'mh-rmail'.

The '+inbox' buffer contains *scan lines*, which are one-line summaries of each incorporated message. You can perform most MH commands on these messages via one- or two-letter commands in either the MH-Folder or MH-Show buffers or by using the 'Message' menu. See `scan`(1) for a description of the contents of the scan lines, and see the Figure in Section 3.2 [Reading Mail Tour], page 8, for an example.

?	Display cheat sheet for the MH-E commands (`mh-help`).
⟨RET⟩	Display message (`mh-show`).
, (comma)	Display message with all header fields (`mh-header-display`).
; (semicolon)	
	Toggle the value of `mh-decode-mime-flag` (`mh-toggle-mh-decode-mime-flag`).
⟨SPC⟩	Display next page in message (`mh-page-msg`).
⟨BS⟩	Display previous page in message (`mh-previous-page`).
>	Append message to end of file (`mh-write-msg-to-file`).
|	Pipe message through shell command (`mh-pipe-msg`).
C-d	Delete range, don't move to next message (`mh-delete-msg-no-motion`).
d	Delete range (`mh-delete-msg`).
D ?	Display cheat sheet for the commands of the current prefix in minibuffer (`mh-prefix-help`).
D ⟨SPC⟩	Display next message in digest (`mh-page-digest`).
D ⟨BS⟩	Display previous message in digest (`mh-page-digest-backwards`).
D b	Break up digest into separate messages (`mh-burst-digest`).
g	Go to a message (`mh-goto-msg`).
k	Delete messages with same subject or thread (`mh-delete-subject-or-thread`).
K ?	Display cheat sheet for the commands of the current prefix in minibuffer (`mh-prefix-help`).

---

[1] If you want to see your old mail as well, use *F r* to pull all your messages into MH-E. Or, give a prefix argument to `mh-rmail` so it will prompt you for folder to visit like *F v* (for example, *C-u M-x mh-rmail* ⟨RET⟩ *bob* ⟨RET⟩). See Chapter 7 [Folders], page 33.

K $\langle\text{TAB}\rangle$ Go to the next button (`mh-next-button`).

K S-$\langle\text{TAB}\rangle$ Go to the previous button (`mh-prev-button`).

K a   Save attachments (`mh-mime-save-parts`).

K e   View attachment externally (`mh-display-with-external-viewer`).

K i   Show attachment verbatim (`mh-folder-inline-mime-part`).

K o   Save (output) attachment (`mh-folder-save-mime-part`).

K t   Toggle option `mh-display-buttons-for-inline-parts-flag` (`mh-toggle-mime-buttons`).

K v   View attachment (`mh-folder-toggle-mime-part`).

M   Edit message (`mh-modify`).

M-<   Display first message (`mh-first-msg`).

M->   Display last message (`mh-last-msg`).

M-n   Display next unread message (`mh-next-unread-msg`).

M-p   Display previous unread message (`mh-previous-unread-msg`).

n   Display next message (`mh-next-undeleted-msg`).

p   Display previous message (`mh-previous-undeleted-msg`).

P ?   Display cheat sheet for the commands of the current prefix in minibuffer (`mh-prefix-help`).

P C   Toggle whether color is used in printing messages (`mh-ps-print-toggle-color`).

P F   Toggle whether printing is done with faces or not (`mh-ps-print-toggle-faces`).

P f   Print range to file (`mh-ps-print-msg-file`).

P l   Print range the old fashioned way (`mh-print-msg`).

P p   Print range (`mh-ps-print-msg`).

X ?   Display cheat sheet for the commands of the current prefix in minibuffer (`mh-prefix-help`).

X s
X u   Unpack message created with `uudecode` or `shar` (`mh-store-msg`).

Mouse-2   Move point to mouse event and show message (`mh-show-mouse`).

Within the MH-Show buffer, the following command is defined.

$\langle\text{RET}\rangle$
Mouse-1
Mouse-2   View contents of button (`mh-press-button`).

The following table lists options in the '`mh-show`' customization group that are used while reading mail.

`mh-bury-show-buffer-flag`
>   On means show buffer is buried (default: 'on').

`mh-clean-message-header-flag`
>   On means remove extraneous header fields (default: 'on').

`mh-decode-mime-flag`
>   On means attachments are handled (default: 'on' if the Gnus 'mm-decode' package is present).

`mh-display-buttons-for-alternatives-flag`
>   On means display buttons for all alternative attachments (default: 'off').

`mh-display-buttons-for-inline-parts-flag`
>   On means display buttons for all inline attachments (default: 'off').

`mh-do-not-confirm-flag`
>   On means non-reversible commands do not prompt for confirmation (default: 'off').

`mh-fetch-x-image-url`
>   Control fetching of 'X-Image-URL:' header field image (default: 'Never Fetch').

`mh-graphical-smileys-flag`
>   On means graphical smileys are displayed (default: 'on').

`mh-graphical-emphasis-flag`
>   On means graphical emphasis is displayed (default: 'on').

`mh-highlight-citation-style`
>   Style for highlighting citations (default: 'Multicolor').

`mh-invisible-header-fields-default`
>   List of hidden header fields (default: a checklist too long to list here).

`mh-invisible-header-fields`
>   Additional header fields to hide (default: `nil`).

`mh-lpr-command-format`
>   Command used to print (default: `"lpr -J '%s'"`).

`mh-max-inline-image-height`
>   Maximum inline image height if 'Content-Disposition:' is not present (default: 0).

`mh-max-inline-image-width`
>   Maximum inline image width if 'Content-Disposition:' is not present(default: 0).

`mh-mhl-format-file`
>   Specifies the format file to pass to the `mhl` program (default: 'Use Default mhl Format (Printing Only)').

`mh-mime-save-parts-default-directory`
>   Default directory to use for *K* a.

`mh-print-background-flag`
> On means messages should be printed in the background (default: 'off').

`mh-show-buffer-mode-line-buffer-id`
> Format string to produce `mode-line-buffer-identification` for show buffers (default: " {show-%s} %d").

`mh-show-maximum-size`
> Maximum size of message (in bytes) to display automatically (default: 0).

`mh-show-use-xface-flag`
> On means display face images in MH-Show buffers (default: 'on').

`mh-store-default-directory`
> Default directory for *X s* (default: 'Current').

`mh-summary-height`
> Number of lines in MH-Folder buffer (including the mode line) (default: depends on size of frame).

The following hooks are available.

`mh-delete-msg-hook`
> Hook run after marking each message for deletion (default: `nil`).

`mh-show-hook`
> Hook run after $\overline{\text{RET}}$ shows a message (default: `nil`).

`mh-show-mode-hook`
> Hook run upon entry to `mh-show-mode` (default: `nil`).

The following faces are available.

`mh-show-cc`
> Face used to highlight 'cc:' header fields.

`mh-show-date`
> Face used to highlight 'Date:' header fields.

`mh-show-from`
> Face used to highlight 'From:' header fields.

`mh-show-header`
> Face used to deemphasize less interesting header fields.

`mh-show-pgg-bad`
> Bad PGG signature face.

`mh-show-pgg-good`
> Good PGG signature face.

`mh-show-pgg-unknown`
> Unknown or untrusted PGG signature face.

`mh-show-signature`
> Signature face.

mh-show-subject
> Face used to highlight 'Subject:' header fields.

mh-show-to
> Face used to highlight 'To:' header fields.

mh-show-xface
> X-Face image face.

The functions and variables introduced here are explained in more detail in the following sections.

6.1 Viewing Your Mail

The command ⟨RET⟩ (mh-show) displays the message that the cursor is on while *Mouse-2* (mh-show-mouse) displays the message that the mouse cursor is on. If the message is already displayed, it scrolls to the beginning of the message. Use ⟨SPC⟩ (mh-page-msg) and ⟨BS⟩ (mh-previous-page) to move forwards and backwards one page at a time through the message. You can give either of these commands a prefix argument that specifies the number of lines to scroll (such as *10* ⟨SPC⟩). The ⟨SPC⟩ command will also show the next undeleted message if it is used at the bottom of a message. MH-E normally hides a lot of the superfluous header fields that mailers add to a message, but if you wish to see all of them, use the command , (comma; mh-header-display).

The option mh-show-maximum-size provides an opportunity to skip over large messages which may be slow to load. The default value of 0 means that all message are shown regardless of size.

A litany of options control what displayed messages look like.

First, the appearance of the header fields can be modified by customizing the associated face: mh-show-to, mh-show-cc, mh-show-from, mh-show-date, and mh-show-subject. The face mh-show-header is used to deemphasize the other, less interesting, header fields.

Normally messages are delivered with a handful of uninteresting header fields. These are hidden by turning on the option mh-clean-message-header-flag (which it is by default). The header fields listed in the option mh-invisible-header-fields-default are hidden, although you can check off any field that you would like to see. Header fields that you would like to hide that aren't listed can be added to the option mh-invisible-header-fields with a couple of caveats. Regular expressions are not allowed. Unique fields should have a ':' suffix; otherwise, the element can be used to render invisible an entire class of fields that start with the same prefix. If you think a header field should be generally ignored, report a bug (see Section C.1 [Bug Reports], page 99).

MH-E can display the content of 'Face:', 'X-Face:', and 'X-Image-URL:' header fields. If any of these fields occur in the header of your message, the sender's face will appear in the 'From:' header field. If more than one of these fields appear, then the first field found in the order 'Face:', 'X-Face:', and 'X-Image-URL:' will be used. The option mh-show-use-xface-flag is used to turn this feature on and off. This feature will be turned on by default if your system supports it.

The first header field used, if present, is the Gnus-specific 'Face:' field[2].

---

[2] The 'Face:' field appeared in GNU Emacs 21 and XEmacs. For more information, see http://quimby.gnus.org/circus/face/.

Next is the traditional 'X-Face:' header field[3]. MH-E renders the foreground and background of the image using the associated attributes of the face mh-show-xface.

Finally, MH-E will display images referenced by the 'X-Image-URL:' header field if neither the 'Face:' nor the 'X-Face:' fields are present[4]. Of the three header fields this is the most efficient in terms of network usage since the image doesn't need to be transmitted with every single mail. The option mh-fetch-x-image-url controls the fetching of the 'X-Image-URL:' header field image with the following values:

'Ask Before Fetching'

You are prompted before the image is fetched. MH-E will remember your reply and will either use the already fetched image the next time the same URL is encountered or silently skip it if you didn't fetch it the first time. This is a good setting.

'Never Fetch'

Images are never fetched and only displayed if they are already present in the cache. This is the default.

There isn't a value of 'Always Fetch' for privacy and DOS (denial of service) reasons. For example, fetching a URL can tip off a spammer that you've read his email (which is why you shouldn't blindly answer yes if you've set this option to 'Ask Before Fetching'). Someone may also flood your network and fill your disk drive by sending a torrent of messages, each specifying a unique URL to a very large file.

The cache of images is found in the directory '.mhe-x-image-cache' within your MH directory. You can add your own face to the 'From:' field too. See Section 9.5 [Picture], page 51.

Normally MH-E takes care of displaying messages itself (rather than calling an MH program to do the work). If you'd rather have mhl display the message (within MH-E), change the option mh-mhl-format-file from its default value of 'Use Default mhl Format (Printing Only)'. You can set this option to 'Use Default mhl Format' to get the same output as you would get if you ran mhl from the shell. If you have a format file that you want MH-E to use, you can set this option to 'Specify an mhl Format File' and enter the name of your format file (mhl(1) or section Using mhl in the MH book tells you how to write one). Your format file should specify a non-zero value for 'overflowoffset' to allow MH-E to parse the header. Note that mhl is always used for printing and forwarding; in this case, the value of mh-mhl-format-file is consulted if you have specified a format file.

If the sender of the message has cited other messages in his message, then MH-E will highlight these citations to emphasize the sender's actual response. The option mh-highlight-citation-style can be customized to change the highlighting style. The 'Multicolor' method uses a different color for each indentation while the 'Monotone' method highlights all citations in red. To disable highlighting of citations entirely, choose 'None'.

Email addresses and URLs in the message are highlighted if the option goto-address-highlight-p is on, which it is by default. To view the web page for a highlighted URL

---

[3] The display of this field requires the uncompface program. Recent versions of XEmacs have internal support for 'X-Face:' images. If your version of XEmacs does not, then you'll need both uncompface and the 'x-face' package.

[4] The display of the images requires the wget program to fetch the image and the convert program from the ImageMagick suite.

or to send a message using a highlighted email address, use *Mouse-2* or `C-c` (RET) (`goto-address-at-point`). See Chapter 8 [Sending Mail], page 39, to see how to configure Emacs to send the message using MH-E.

It is a long standing custom to inject body language using a cornucopia of punctuation, also known as the *smileys*. MH-E can render these as graphical widgets if the option `mh-graphical-smileys-flag` is turned on, which it is by default. Smileys include patterns such as :-) and ;-). Similarly, a few typesetting features are indicated in ASCII text with certain characters. If your terminal supports it, MH-E can render these typesetting directives naturally if the option `mh-graphical-emphasis-flag` is turned on, which it is by default. For example, _underline_ will be underlined, *bold* will appear in **bold**, /italics/ will appear in *italics*, and so on. See the option `gnus-emphasis-alist` for the whole list. Both of these options are disabled if the option `mh-decode-mime-flag` is turned off. See Section 6.2 [Viewing Attachments], page 23.

MH-E normally renders signatures and vCards in italics so that the body of the message stands out more. MH-E depends on the presence of the *signature separator* (`"-- "`) to do this. You can also customize the face `mh-show-signature` so the appearance of the signature block is more to your liking.

Two hooks can be used to control how messages are displayed. The first hook, `mh-show-mode-hook`, is called early on in the process of the message display. It is usually used to perform some action on the message's content. The second hook, `mh-show-hook`, is the last thing called after messages are displayed. It's used to affect the behavior of MH-E in general or when `mh-show-mode-hook` is too early.

For those who like to modify their mode lines, use `mh-show-buffer-mode-line-buffer-id` to modify the mode line in the MH-Show buffers. Place the two escape strings '`%s`' and '`%d`', which will display the folder name and the message number, respectively, somewhere in the string in that order. The default value of `"{show-%s} %d"` yields a mode line of

```
-----{show-+inbox} 4      (MH-Show)--Bot--------------------------------
```

6.2 Viewing Attachments

MH has the ability to display MIME (Multipurpose Internet Mail Extensions) messages which are simply messages with additional *body parts* or *attachments*. You can use the MH commands `show`[5] or `mhshow`[6] from the shell to read MIME messages[7].

MH-E can handle attachments as well if the Gnus '`mm-decode`' package is present. If so, the option `mh-decode-mime-flag` will be on. Otherwise, you'll see the MIME body parts rather than text or attachments. There isn't much point in turning off the option `mh-decode-mime-flag`; however, you can inspect it if it appears that the body parts are not being interpreted correctly or toggle it with the command `;` (semicolon; `mh-toggle-mh-decode-mime-flag`) to view the raw message. This option also controls the display of quoted-printable messages and other graphical widgets. See Section 6.1 [Viewing], page 21.

Attachments in MH-E are indicated by *buttons* like this:

---

[5] See the section Reading Mail: inc show next prev in the MH book.

[6] See the section Reading MIME Mail in the MH book.

[7] You can call them directly from Emacs if you're running the X Window System: type *M-! xterm -e mhshow message-number*. You can leave out the '`xterm -e`' if you use `mhlist` or `mhstore`.

```
[1. image/jpeg; foo.jpg]...
```

To view the contents of the button, use either *Mouse-1* or *Mouse-2* on the button or
⟨RET⟩ (`mh-press-button`) when the cursor is over the button. This command is a toggle so
if you use it again on the same attachment, it is hidden. If Emacs does not know how to
display the attachment, then Emacs offers to save the attachment in a file. To move the
cursor to the next button, use the command *K* ⟨TAB⟩ (`mh-next-button`). If the end of the
buffer is reached then the search wraps over to the start of the buffer. To move the cursor
to the previous button, use the command *K S-*⟨TAB⟩ (`mh-prev-button`). If the beginning of
the buffer is reached then the search wraps over to the end of the buffer.

Another way to view the contents of a button is to use the command *K v* (`mh-folder-`
`toggle-mime-part`). This command displays (or hides) the attachment associated with the
button under the cursor. If the cursor is not located over a button, then the cursor first
moves to the next button, wrapping to the beginning of the message if necessary. This
command has the advantage over the previous commands of working from the MH-Folder
buffer. You can also provide a numeric prefix argument (as in *4 K v*) to view the attachment
labeled with that number. If Emacs does not know how to display the attachment, then
Emacs offers to save the attachment in a file.

If Emacs does not know how to view an attachment, you could save it into a file and then
run some program to open it. It is easier, however, to launch the program directly from
MH-E with the command *K e* (`mh-display-with-external-viewer`). While you'll most
likely use this to view spreadsheets and documents, it is also useful to use your browser to
view HTML attachments with higher fidelity than what Emacs can provide. This command
displays the attachment associated with the button under the cursor. If the cursor is not
located over a button, then the cursor first moves to the next button, wrapping to the
beginning of the message if necessary. You can provide a numeric prefix argument (as in *4
K e*) to view the attachment labeled with that number. This command tries to provide a
reasonable default for the viewer by calling the Emacs function `mailcap-mime-info`. This
function usually reads the file '`/etc/mailcap`'.

Use the command *K o* (`mh-folder-save-mime-part`) to save attachments (the mnemonic
is "output"). This command saves the attachment associated with the button under the
cursor. If the cursor is not located over a button, then the cursor first moves to the next
button, wrapping to the beginning of the message if necessary. You can also provide a
numeric prefix argument (as in *3 K o*) to save the attachment labeled with that number.
This command prompts you for a filename and suggests a specific name if it is available.

You can save all of the attachments at once with the command *K a* (`mh-mime-save-`
`parts`). The attachments are saved in the directory specified by the option `mh-mime-save-`
`parts-default-directory` unless you use a prefix argument (as in *C-u K a*) in which case
you are prompted for the directory. These directories may be superseded by MH profile
components, since this function calls on `mhstore` (`mhn`) to do the work.

The default value for the option `mh-mime-save-parts-default-directory` is '`Prompt`
`Always`' so that you are always prompted for the directory in which to save the attachments.
However, if you usually use the same directory within a session, then you can set this option
to '`Prompt the First Time`' to avoid the prompt each time. you can make this directory
permanent by choosing '`Directory`' and entering the directory's name.

The sender can request that attachments should be viewed inline so that they do not really appear like an attachment at all to the reader. Most of the time, this is desirable, so by default MH-E suppresses the buttons for inline attachments. On the other hand, you may receive code or HTML which the sender has added to his message as inline attachments so that you can read them in MH-E. In this case, it is useful to see the buttons so that you know you don't have to cut and paste the code into a file; you can simply save the attachment. If you want to make the buttons visible for inline attachments, you can use the command `K t` (`mh-toggle-mime-buttons`) to toggle the visibility of these buttons. You can turn on these buttons permanently by turning on the option `mh-display-buttons-for-inline-parts-flag`.

MH-E cannot display all attachments inline however. It can display text (including HTML) and images.

Some older mail programs do not insert the needed plumbing[8] to tell MH-E whether to display the attachments inline or not. If this is the case, MH-E will display these images inline if they are smaller than the window. However, you might want to allow larger images to be displayed inline. To do this, you can change the options `mh-max-inline-image-width` and `mh-max-inline-image-height` from their default value of zero to a large number. The size of your screen is a good choice for these numbers.

Sometimes, a mail program will produce multiple alternatives of an attachment in increasing degree of faithfulness to the original content. By default, only the preferred alternative is displayed. If the option `mh-display-buttons-for-alternatives-flag` is on, then the preferred part is shown inline and buttons are shown for each of the other alternatives.

Many people prefer to see the 'text/plain' alternative rather than the 'text/html' alternative. To do this in MH-E, customize the option `mm-discouraged-alternatives`, and add 'text/html'. The next best alternative, if any, will be shown.

You can view the raw contents of an attachment with the command `K i` (`mh-folder-inline-mime-part`). This command displays (or hides) the contents of the attachment associated with the button under the cursor verbatim. If the cursor is not located over a button, then the cursor first moves to the next button, wrapping to the beginning of the message if necessary. You can also provide a numeric prefix argument (as in `4 K i`) to view the attachment labeled with that number.

For additional information on buttons, see the chapters Article Buttons and MIME Commands in the *The Gnus Manual*.

6.3 HTML

MH-E can display messages that have been sent in HTML[9]. The content of the message will appear in the MH-Show buffer as you would expect if the entire message is HTML, or there is an inline HTML body part. However, if there is an HTML body part that is an attachment, then you'll see a button like this:

        [1. text/html; foo.html]...

To see how to read the contents of this body part, see Section 6.2 [Viewing Attachments], page 23.

---

[8] This plumbing is the 'Content-Disposition:' header field.

[9] This feature depends on a version of Gnus that is at least 5.10.

The browser that MH-E uses is determined by the option `mm-text-html-renderer`. The default setting is set automatically based upon the presence of a known browser on your system. If you wish to use a different browser, then set this option accordingly. See the documentation for the browser you use for additional information on how to use it. In particular, find and disable the option to render images as this can tip off spammers that the email address they have used is valid.

If you're confused about which `mm-text-html-renderer` to use, here's a brief description of each, sorted by popularity, that includes the results of a quick poll of MH-E users from 2005-12-23.

'`w3m`' 7   The '`w3m`' browser requires an external program. It's quick, produces pretty nice output, and best of all, it's the only browser that highlights links. These can be clicked with *Mouse-2* to view the content of the link in '`w3m`' or with *S-Mouse-2* to view the content of the link in an external browser. The '`w3m`' browser handles tables well and actually respects the table's width parameter (which can cause text to wrap if the author didn't anticipate that the page would be viewed in Emacs).

'`w3m-standalone`' 3
            This browser, along with '`nil`' for the external browser, are the only choices that work without having to download a separate lisp package or external program. This browser is quick, but does not show links. It handles simple tables but some tables get rendered much wider than the Emacs frame. This browser was the only one not to handle the escape '`–`' (it printed a '?'), but it did render '`®`'.

'`links`' 1   The '`links`' browser requires an external program. It's quick, and produces nicer output than '`lynx`' on single column mails in tables. However, it doesn't show links and it doesn't do as nice a job on multi-column tables as some lines wrap. At least it fits in 80 columns and thus seems better than '`w3`' and '`w3m-standalone`'. Converts escapes such as '`®`' to (R).

'`lynx`' 1   The '`lynx`' browser requires an external program. It's quick and produces pretty decent output but it doesn't show links. It doesn't seem to do multi-column tables which makes output much cleaner. It centers the output and wraps long lines more than most. Handles '`®`'.

'`nil`' 1   This choice obviously requires an external browser. Like '`w3m-standalone`', it works out of the box. With this setting, HTML messages have a button for the body part which you can view with *K v* (`mh-folder-toggle-mime-part`).

'`w3`' 0   This choice does not require an external program as all of the rendering is done in lisp. You do need to get the package separately. This browser is **slow**, and doesn't appear to have been updated since 2001 and the author hasn't responded to my emails. It displays unknown tags instead of hiding them, so you get to see all the Microsoft crap in certain messages. Tends to make multi-column tables wider than even a full-screen Emacs can handle. Like '`w3m`', you can follow links, but you have to find them first as they are not highlighted. Performs well on single-column tables and handles escapes such as '`®`'.

'`html2text`' 0

> The '`html2text`' browser requires an external program. I noticed that it can do some nasty things with simple HTML mails (like filling the entire message as if it were one paragraph, including signature). On another message, it displayed half of the HTML tags for some reason.

For a couple more sources of information about `mm-text-html-renderer`, see section Display Customization in the *The Emacs MIME Manual* and the the documentation for the Gnus command `W h` (see section Article Washing in the *The Gnus Manual*).

6.4 Digests

A digest is a message that contains other messages. Special MH-E commands let you read digests conveniently. You can use ⟨SPC⟩ and ⟨BS⟩ to page through the digest as if it were a normal message, but if you wish to skip to the next message in the digest, use `D` ⟨SPC⟩ (`mh-page-digest`). To return to a previous message, use `D` ⟨BS⟩ (`mh-page-digest-backwards`).

Another handy command is `D b` (`mh-burst-digest`). This command uses the MH command `burst`[10] to break out each message in the digest into its own message. Using this command, you can quickly delete unwanted messages, like this: Once the digest is split up, toggle out of MH-Folder Show mode with `t` (see Chapter 7 [Folders], page 33) so that the scan lines fill the screen and messages aren't displayed. Then use `d` (see Chapter 6 [Reading Mail], page 17) to quickly delete messages that you don't want to read (based on the '`Subject:`' header field). You can also burst the digest to reply directly to the people who posted the messages in the digest. One problem you may encounter is that the '`From:`' header fields are preceded with a '`>`' so that your reply can't create the '`To:`' field correctly. In this case, you must correct the '`To:`' field yourself. This is described later (see Chapter 9 [Editing Drafts], page 45).

6.5 Signed and Encrypted Messages

You can read encrypted or signed PGP or GPG messages with MH-E[11]. This section assumes that you already have a good understanding of GPG and have set up your keys appropriately.

If someone sends you a signed message, here is what you'll see:

```
[[PGP Signed Part:Bill Wohler <wohler@stop.mail-abuse.org>]]
This is a signed message.

[[End of PGP Signed Part]]
```

If the key for the given signature is not in your keychain, you'll be given the opportunity to fetch the key from a key server and verify the key. If the message is really large, the verification process can take a long time. You can press `C-g` at any time to cancel[12].

If the signature doesn't check out, you might see something like this:

---

[10] See the section Bursting Messages in the MH book.

[11] This feature depends on post-5.10 versions of Gnus. *MIME Security with OpenPGP* is documented in RFC 3156. However, MH-E can also decrypt old-style PGP messages that are not in MIME format.

[12] Unfortunately in the current version, the validation process doesn't display a message so it appears that MH-E has hung. We hope that this will be fixed in the future.

```
[[PGP Signed Part:Failed]]
This is a signed message.
This is garbage added after the signature was made.

[[End of PGP Signed Part]]
```

If someone sends you an encrypted message, MH-E will ask for your passphrase to decrypt the message. You should see something like this:

```
[[PGP Encrypted Part:OK]]

[[PGP Signed Part:Bill Wohler <wohler@stop.mail-abuse.org>]]
This is the secret message.

[[End of PGP Signed Part]]

[[End of PGP Encrypted Part]]
```

If there is a problem decrypting the message, the button will say:

```
[[PGP Encrypted Part:Failed]]
```

You can read the contents of this button using the methods described in Section 6.2 [Viewing Attachments], page 23. If the message were corrupted, you'd see this:

```
[[PGP Encrypted Part:Failed]
Invalid base64 data]
```

If your passphrase were incorrect, you'd see something like this:

```
[GNUPG:] ENC_TO CD9C88BB610BD9AD 1 0
[GNUPG:] USERID_HINT CD9C88BB610BD9AD Bill Wohler <wohler@stop.mail-abuse.org>
[GNUPG:] NEED_PASSPHRASE CD9C88BB610BD9AD CD9C88BB610BD9AD 1 0
[GNUPG:] BAD_PASSPHRASE CD9C88BB610BD9AD
gpg: encrypted with 1024-bit RSA key, ID 610BD9AD, created 1997-09-09
      "Bill Wohler <wohler@stop.mail-abuse.org>"
gpg: public key decryption failed: bad passphrase
[GNUPG:] BEGIN_DECRYPTION
[GNUPG:] DECRYPTION_FAILED
gpg: decryption failed: secret key not available
[GNUPG:] END_DECRYPTION

gpg exited abnormally: '2'
```

The appearance of the buttons is controlled by the faces `mh-show-pgg-good`, `mh-show-pgg-bad`, and `mh-show-pgg-unknown` depending on the validity of the signature. The latter is used whether the signature is unknown or untrusted.

The 'pgg' customization group may have some settings which may interest you. See *The PGG Manual*.

6.6 Printing Your Mail

To print messages in MH-E, use the command *P p* (`mh-ps-print-msg`). You can print all the messages in a range (as in *C-u P p 1 3 5-7 last:5 frombob* $\overline{\text{RET}}$, see Section 4.2 [Ranges], page 12). You can also send the output to a file with *P f* (`mh-ps-print-msg-file`). This command will print inline text attachments but will not decrypt messages. However, when a message is displayed in an MH-Show buffer, then that buffer is used verbatim for printing with the caveat that only text attachments, if opened inline, are printed. Therefore, encrypted messages can be printed by showing and decrypting them

first. The commands *P p* and *P f* do not use the options `mh-lpr-command-format` or `mh-print-background-flag`, described below.

Colors are emulated on black-and-white printers with shades of gray. This might produce illegible output, even if your screen colors only use shades of gray. If this is the case, try using the command *P C* (`mh-ps-print-toggle-color`) to toggle between color, no color, and a black and white representation of the colors and see which works best. You change this setting permanently by customizing the option `ps-print-color-p`.

Another related function is the command *P F* (`mh-ps-print-toggle-faces`). This command toggles between using faces and not. When faces are enabled, the printed message will look very similar to the message in the MH-Show buffer.

MH-E uses the '`ps-print`' package to do the printing, so you can customize the printing further by going to the '`ps-print`' customization group.

An alternative to using the '`ps-print`' package is the command *P l* (`mh-print-msg`) (the *l* is for *l*ine printer or *l*pr). You can print all the messages in a range. The message is formatted with `mhl`[13] and printed with the `lpr` command.

The command *P l* uses two options. The option `mh-lpr-command-format` contains the Unix command line which performs the actual printing. The string can contain one escape, '`%s`', which is replaced by the name of the folder and the message number and is useful for print job names. The default setting is `"lpr -J '%s'"`. I use `"mpage -h'%s' -b Letter -H1of -mlrtb -P"` which produces a nice header and adds a bit of margin so the text fits within my printer's margins. Normally messages are printed in the foreground. If this is slow on your system, you may elect to turn on the option `mh-print-background-flag` to print in the background. If you do this, do not delete the message until it is printed or else the output may be truncated. These options are not used by the commands *P p* or *P f*.

6.7 Files and Pipes

MH-E does offer a couple of commands that are not a part of MH. The first one, `>` (`mh-write-msg-to-file`), writes a message to a file. You are prompted for the filename. If the file already exists, the message is appended to it. You can also write the message to the file without the header by specifying a prefix argument (such as *C-u >* `/tmp/foobar` ⟨RET⟩). Subsequent writes to the same file can be made with the command *!* (`mh-refile-or-write-again`).

You can also pipe the message through a Unix shell command with the command *|* (`mh-pipe-msg`). You are prompted for the Unix command through which you wish to run your message. If you give a prefix argument to this command, the message header is included in the text passed to the command (the contrived example *C-u |* `lpr` would be done with the *l* command instead).

If the message is a shell archive `shar` or has been run through `uuencode` use *X s* (`mh-store-msg`) to extract the body of the message. The default directory for extraction is the current directory; however, you have a chance to specify a different extraction directory. The next time you use this command, the default directory is the last directory you used. If you would like to change the initial default directory, customize the option `mh-store-`

---

[13] See the section Using mhl in the MH book.

`default-directory`, change the value from 'Current' to 'Directory', and then enter the name of the directory for storing the content of these messages.

By the way, `X s` calls the Emacs Lisp function `mh-store-buffer`. I mention this because you can use it directly if you're editing a buffer that contains a file that has been run through `uuencode` or `shar`. For example, you can extract the contents of the current buffer in your home directory by typing `M-x mh-store-buffer` (RET) ~ (RET).

6.8 Navigating

To move on to the next message, use the command `n` (`mh-next-undeleted-msg`); use `p` (`mh-previous-undeleted-msg`) to read the previous message. To move to the next unread message, use `M-n` (`mh-next-unread-msg`); use `M-p` (`mh-previous-unread-msg`) to move to the previous unread message. These commands can be given a prefix argument to specify how many messages to skip (for example, `5 n`). You can also move to a specific message with `g` (`mh-goto-msg`). You can enter the message number either before or after typing `g`. In the latter case, Emacs prompts you. Finally, you can go to the first or last message with `M-<` (`mh-first-msg`) and `M->` (`mh-last-msg`) respectively.

You can also use the Emacs commands `C-p` (`previous-line`) and `C-n` (`next-line`) to move up and down the scan lines in the MH-Folder window. These commands can be used in conjunction with (RET) to look at deleted or refiled messages.

To mark a message for deletion, use the command `d` (`mh-delete-msg`). A 'D' is placed by the message in the scan window, and the next undeleted message is displayed. If the previous command had been `p`, then the next message displayed is the first undeleted message previous to the message just deleted. Use `n` to force subsequent `d` commands to move forward to the next undeleted message after deleting the message under the cursor. You may also specify a range (for example, `C-u d 1 3 5-7 last:5 frombob` (RET), see Section 4.2 [Ranges], page 12).

The command `C-d` (`mh-delete-msg-no-motion`) marks the message (or messages in range) for deletion but leaves the cursor at the current message in case you wish to perform other operations on the message.

And to delete more messages faster, you can use `k` (`mh-delete-subject-or-thread`) to delete all the messages with the same subject as the current message. This command puts these messages in a sequence named 'subject'. You can undo this action by using `u` (`mh-undo`) with a prefix argument and then specifying the 'subject' sequence. However, if the buffer is displaying a threaded view of the folder then `k` behaves like `T d` (`mh-thread-delete`). See Chapter 16 [Threading], page 77.

However you mark a message for deletion, the command `x` (`mh-execute-commands`) actually carries out the deletion (see Chapter 7 [Folders], page 33).

The hook `mh-delete-msg-hook` is called after you mark a message for deletion. For example, a past maintainer of MH-E used this once when he kept statistics on his mail usage.

6.9 Miscellaneous Commands and Options

This section contains a few more miscellaneous commands and options.

There are times when you need to edit a message. For example, you may need to fix a broken Content-Type header field. You can do this with the command `M` (`mh-modify`). It displays the raw message in an editable buffer. When you are done editing, save and kill the buffer as you would any other.

Commands such as `mh-pack-folder` prompt to confirm whether to process outstanding moves and deletes or not before continuing. Turning on the option `mh-do-not-confirm-flag` means that these actions will be performed—which is usually desired but cannot be retracted—without question[14].

The option `mh-summary-height` controls the number of scan lines displayed in the MH-Folder window, including the mode line. The default value of this option is '`Automatic`' which means that the MH-Folder buffer will maintain the same proportional size if the frame is resized. If you'd prefer a fixed height, then choose the '`Fixed Size`' option and enter the number of lines you'd like to see.

Normally the buffer for displaying messages is buried at the bottom at the buffer stack. You may wish to disable this feature by turning off the option `mh-bury-show-buffer-flag`. One advantage of not burying the show buffer is that one can delete the show buffer more easily in an electric buffer list because of its proximity to its associated MH-Folder buffer. Try running `M-x electric-buffer-list` to see what I mean.

Before we leave this section, I'll include a function that I use as a front end to MH-E[15]. It toggles between your working window configuration, which may be quite involved—windows filled with source, compilation output, man pages, and other documentation—and your MH-E window configuration. Like the rest of the customization described in this section, simply add the following code to '`~/.emacs`'.

---

[14] In previous versions of MH-E, this option suppressed the confirmation in `mh-kill-folder`. Since this kept most users from setting this option, `mh-kill-folder` was modified in version 6.0 to always ask for confirmation subject to `mh-kill-folder-suppress-prompt-hook`. See Chapter 7 [Folders], page 33.

[15] Stephen Gildea's favorite binding is `(global-set-key "\C-cr" 'mh-rmail)`.

```
(defvar my-mh-screen-saved nil
  "Set to non-nil when MH-E window configuration shown.")
(defvar my-normal-screen nil "Normal window configuration.")
(defvar my-mh-screen nil "MH-E window configuration.")

(defun my-mh-rmail (&optional arg)
  "Toggle between MH-E and normal screen configurations.
With non-nil or prefix argument, inc mailbox as well
when going into mail."
  (interactive "P")                    ; user callable function, P=prefix arg
  (setq my-mh-screen-saved             ; save state
        (cond
         ;; Bring up MH-E screen if arg or normal window configuration.
         ;; If arg or +inbox buffer doesn't exist, run mh-rmail.
         ((or arg (null my-mh-screen-saved))
          (setq my-normal-screen (current-window-configuration))
          (if (or arg (null (get-buffer "+inbox")))
              (mh-rmail)
            (set-window-configuration my-mh-screen))
          t)                           ; set my-mh-screen-saved to t
         ;; Otherwise, save MH-E screen and restore normal screen.
         (t
          (setq my-mh-screen (current-window-configuration))
          (set-window-configuration my-normal-screen)
          nil))))                      ; set my-mh-screen-saved to nil

(global-set-key "\C-x\r" 'my-mh-rmail)   ; call with C-x RET
```

Starting MH-E

If you type an argument (*C-u*) or if `my-mh-screen-saved` is `nil` (meaning a non-MH-E window configuration), the current window configuration is saved, either the '`+inbox`' buffer is displayed or `mh-rmail` is run, and the MH-E window configuration is shown. Otherwise, the MH-E window configuration is saved and the original configuration is displayed.

7 Organizing Your Mail with Folders

This chapter discusses the things you can do with folders within MH-E. The commands in
this chapter are also found in the 'Folder' and 'Message' menus.

`?`	Display cheat sheet for the MH-E commands (`mh-help`).
`!`	Repeat last output command (`mh-refile-or-write-again`).
`c`	Copy range to folder (`mh-copy-msg`).
`F ?`	Display cheat sheet for the commands of the current prefix in minibuffer (`mh-prefix-help`).
`F '`	Display ticked messages (`mh-index-ticked-messages`).
`F c`	Delete range from the 'unseen' sequence (`mh-catchup`).
`F k`	Remove folder (`mh-kill-folder`).
`F l`	List all folders (`mh-list-folders`).
`F n`	Display unseen messages (`mh-index-new-messages`).
`F p`	Pack folder (`mh-pack-folder`).
`F q`	Display messages in any sequence (`mh-index-sequenced-messages`).
`F r`	Rescan folder (`mh-rescan-folder`).
`F s`	Search your MH mail (`mh-search`).
`F S`	Sort folder (`mh-sort-folder`).
`F u`	Undo all refiles and deletes in the current folder (`mh-undo-folder`).
`F v`	Visit folder (`mh-visit-folder`).
`o`	Refile (output) range into folder (`mh-refile-msg`).
`q`	Quit the current MH-E folder (`mh-quit`).
`t`	Toggle between MH-Folder and MH-Folder Show modes (`mh-toggle-showing`).
`u`	Undo pending deletes or refiles in range (`mh-undo`).
`x`	Process outstanding delete and refile requests (`mh-execute-commands`).

The 'mh-folder' customization group is used to tune these commands.

mh-new-messages-folders
: Folders searched for the 'unseen' sequence (default: `Inbox`).

mh-ticked-messages-folders
: Folders searched for `mh-tick-seq` (default: `t`).

mh-large-folder
: The number of messages that indicates a large folder (default: 200).

mh-recenter-summary-flag
: On means to recenter the summary window (default: 'off').

`mh-recursive-folders-flag`
> On means that commands which operate on folders do so recursively (default: `off`).

`mh-sortm-args`
> Additional arguments for `sortm` (default: `nil`).

The following hooks are available.

`mh-after-commands-processed-hook`
> Hook run by `x` after performing outstanding refile and delete requests (default: `nil`).

`mh-before-commands-processed-hook`
> Hook run by `x` before performing outstanding refile and delete requests (default: `nil`).

`mh-before-quit-hook`
> Hook run by q before quitting MH-E (default: `nil`).

`mh-folder-mode-hook`
> Hook run by `mh-folder-mode` when visiting a new folder (default: `nil`).

`mh-kill-folder-suppress-prompt-hook`
> Abnormal hook run at the beginning of `mh-kill-folder` (default: `'mh-search-p`).

`mh-quit-hook`
> Hook run by q after quitting MH-E (default: `nil`).

`mh-refile-msg-hook`
> Hook run by o after marking each message for refiling (default: `nil`).

The following faces are available for customizing the appearance of the MH-Folder buffer. See [Appendix A [Scan Line Formats], page 91](#).

`mh-folder-address`
> Recipient face.

`mh-folder-body`
> Body text face.

`mh-folder-cur-msg-number`
> Current message number face.

`mh-folder-date`
> Date face.

`mh-folder-deleted`
> Deleted message face.

`mh-folder-followup`
> `Re:` face.

`mh-folder-msg-number`
> Message number face.

`mh-folder-refiled`

> Refiled message face.

`mh-folder-sent-to-me-hint`

> Fontification hint face in messages sent directly to us. The detection of messages sent to us is governed by the scan format `mh-scan-format-nmh` and regular expression `mh-scan-sent-to-me-sender-regexp`.

`mh-folder-scan-format`

> Sender face in messages sent directly to us. The detection of messages sent to us is governed by the scan format `mh-scan-format-nmh` and regular expression `mh-scan-sent-to-me-sender-regexp`.

`mh-folder-subject`

> Subject face.

`mh-folder-tick`

> Ticked message face.

`mh-folder-to`

> 'To:' face.

The hook `mh-folder-mode-hook` is called when visiting a new folder in MH-Folder mode. This could be used to set your own key bindings, for example:

```
(defvar my-mh-init-done nil
  "Non-nil when one-time MH-E settings made.")

(defun my-mh-folder-mode-hook ()
  "Hook to set key bindings in MH-Folder mode."
  (if (not my-mh-init-done)              ; only need to bind the keys once
      (progn
        (local-set-key "//" 'my-search-msg)
        (local-set-key "b" 'mh-burst-digest)    ; better use of b
        (setq my-mh-init-done t))))

(add-hook 'mh-folder-mode-hook 'my-mh-folder-mode-hook)

(defun my-search-msg ()
  "Search for a regexp in the current message."
  (interactive)                          ; user function
  (save-window-excursion
    (other-window 1)                     ; go to next window
    (isearch-forward-regexp)))           ; string search; hit return
                                         ;   when done
```

Create additional key bindings via mh-folder-mode-hook

MH-E has analogies for each of the MH **folder** and **refile** commands[1]. To refile a message in another folder, use the command *o* (`mh-refile-msg`) (mnemonic: "output"). You are prompted for the folder name (see Section 4.3 [Folder Selection], page 13). Note that this command can also be used to create folders. If you specify a folder that does not exist, you will be prompted to create it. The hook `mh-refile-msg-hook` is called after a message is marked to be refiled.

---

[1] See the sections Your Current Folder: folder and Moving and Linking Messages: refile in the MH book.

If you are refiling several messages into the same folder, you can use the command *!*
(`mh-refile-or-write-again`) to repeat the last refile or write (for the description of *>*
(`mh-write-msg-to-file`), see Section 6.7 [Files and Pipes], page 29). You can use a range
in either case (for example, *C-u o 1 3 5-7 last:5 frombob* $\overline{\text{RET}}$, see Section 4.2 [Ranges],
page 12).

If you've deleted a message or refiled it, but changed your mind, you can cancel the
action before you've executed it. Use *u* (`mh-undo`) to undo a refile on or deletion of a single
message. You can also undo refiles and deletes for messages that are found in a given range
(see Section 4.2 [Ranges], page 12).

Alternatively, you can use *F u* (`mh-undo-folder`) to undo all refiles and deletes in the
current folder.

If you've marked messages to be deleted or refiled and you want to go ahead and delete
or refile the messages, use *x* (`mh-execute-commands`). Many MH-E commands that may
affect the numbering of the messages (such as *F r* or *F p*) will ask if you want to process
refiles or deletes first and then either run *x* for you or undo the pending refiles and deletes.

The command *x* runs `mh-before-commands-processed-hook` before the commands are
processed and `mh-after-commands-processed-hook` after the commands are processed.
Variables that are useful with the former hook include `mh-delete-list` and `mh-refile-`
`list` which can be used to see which changes will be made to the current folder, `mh-`
`current-folder`. Variables that are useful with the latter hook include `mh-folders-`
`changed`, which lists which folders were affected by deletes and refiles. This list will always
include the current folder `mh-current-folder`.

If you wish to copy a message to another folder, you can use the command *c* (`mh-copy-`
`msg`) (see the '`-link`' argument to `refile(1)`). Like the command *o*, this command prompts
you for the name of the target folder and you can specify a range (see Section 4.2 [Ranges],
page 12). Note that unlike the command *o*, the copy takes place immediately. The original
copy remains in the current folder.

The command *t* (`mh-toggle-showing`) switches between MH-Folder mode and MH-
Folder Show mode[2]. MH-Folder mode turns off the associated show buffer so that you can
perform operations on the messages quickly without reading them. This is an excellent
way to prune out your junk mail or to refile a group of messages to another folder for later
examination.

When you use *t* to toggle from MH-Folder Show mode to MH-Folder mode, the MH-Show
buffer is hidden and the MH-Folder buffer is left alone. Setting `mh-recenter-summary-`
`flag` to a non-`nil` value causes the toggle to display as many scan lines as possible, with
the cursor at the middle. The effect of `mh-recenter-summary-flag` is rather useful, but it
can be annoying on a slow network connection.

When you want to read the messages that you have refiled into folders, use the command
F v (`mh-visit-folder`) to visit the folder. You are prompted for the folder name. The
folder buffer will show just unseen messages if there are any; otherwise, it will show all the
messages in the buffer as long there are fewer than `mh-large-folder` messages. If there
are more, then you are prompted for a range of messages to scan. You can provide a prefix
argument in order to specify a range of messages to show when you visit the folder (see

---

[2] For you Emacs wizards, this is implemented as an Emacs minor mode.

Section 4.2 [Ranges], page 12). In this case, regions are not used to specify the range and `mh-large-folder` is ignored. Note that this command can also be used to create folders. If you specify a folder that does not exist, you will be prompted to create it.

If you forget where you've refiled your messages, you can find them using *F s* (`mh-search`). See Chapter 15 [Searching], page 71.

If you use a program such as `procmail` to file your incoming mail automatically, you can display new, unseen, messages using the command *F n* (`mh-index-new-messages`). All messages in the '`unseen`' sequence from the folders in `mh-new-messages-folders` are listed. However, this list of folders can be overridden with a prefix argument: with a prefix argument, enter a space-separated list of folders, or nothing to search all folders.

If you have ticked messages (see Chapter 18 [Sequences], page 81), you can display them using the command *F '* (`mh-index-ticked-messages`). All messages in the '`tick`' sequence from the folders in `mh-ticked-messages-folders` are listed. With a prefix argument, enter a space-separated list of folders, or nothing to search all folders.

You can display messages in any sequence with the command *F q* (`mh-index-sequenced-messages`). All messages from the folders in `mh-new-messages-folders` in the sequence you provide are listed. With a prefix argument, enter a space-separated list of folders at the prompt, or nothing to search all folders.

Set the options `mh-new-messages-folders` and `mh-ticked-messages-folders` to '`Inbox`' to search the '`+inbox`' folder or '`All`' to search all of the top level folders. Otherwise, list the folders that should be searched with the '`Choose Folders`' menu item. See `mh-recursive-folders-flag`.

Other commands you can perform on folders include: *F l* (`mh-list-folders`), to place a listing of all the folders in your mail directory in a buffer called '`*MH-E Folders*`' (see Chapter 20 [Miscellaneous], page 89); *F k* (`mh-kill-folder`), to remove a folder; *F S* (`mh-sort-folder`), to sort the messages by date (see `sortm`(1) to see how to sort by other criteria); *F p* (`mh-pack-folder`), to pack a folder, removing gaps from the numbering sequence; and *F r* (`mh-rescan-folder`), to rescan the folder, which is useful to grab all messages in your '`+inbox`' after processing your new mail for the first time. If you don't want to rescan the entire folder, the commands *F r* or *F p* will accept a range (see Section 4.2 [Ranges], page 12).

By default, operations on folders work only one level at a time. Set `mh-recursive-folders-flag` to non-`nil` to operate on all folders. This mostly means that you'll be able to see all your folders when you press ⒯ⒶⒷ when prompted for a folder name.

The hook `mh-kill-folder-suppress-prompt-hooks` is an abnormal hook run at the beginning of the command *k*. The hook functions are called with no arguments and should return a non-nil value to suppress the normal prompt when you remove a folder. This is useful for folders that are easily regenerated. The default value of `mh-search-p` suppresses the prompt on folders generated by searching.

NOTE

Use this hook with care. If there is a bug in your hook which returns `t` on '`+inbox`' and you press *k* by accident in the `+inbox` folder, you will not be happy.

The option `mh-sortm-args` holds extra arguments to pass on to the command `sortm`[3] when a prefix argument is used with *F S*. Normally default arguments to `sortm` are specified in the MH profile. This option may be used to provide an alternate view. For example, ''`(\"-nolimit\" \"-textfield\" \"subject\")`' is a useful setting.

When you want to quit using MH-E and go back to editing, you can use the *q* (`mh-quit`) command. This buries the buffers of the current MH-E folder and restores the buffers that were present when you first ran *M-x mh-rmail*. It also removes any MH-E working buffers whose name begins with ' `*mh-`' or '`*MH-E `' (see Chapter 20 [Miscellaneous], page 89). You can later restore your MH-E session by selecting the '`+inbox`' buffer or by running *M-x mh-rmail* again.

The two hooks `mh-before-quit-hook` and `mh-quit-hook` are called by *q*. The former one is called before the quit occurs, so you might use it to perform any MH-E operations; you could perform some query and abort the quit or call `mh-execute-commands`, for example. The latter is not run in an MH-E context, so you might use it to modify the window setup. If you find that *q* buries a lot of buffers that you would rather remove, you can use both `mh-before-quit-hook` and `mh-quit-hook` to accomplish that.

```
(defvar my-mh-folder-buffer-to-delete nil
  "Folder buffer that is being quit.")

(defun my-mh-before-quit-hook ()
  "Save folder buffer that is to be deleted."
  (setq my-mh-folder-buffer-to-delete (current-buffer)))

(defun my-mh-quit-hook ()
  "Kill folder buffer rather than just bury it."
  (set-buffer my-mh-folder-buffer-to-delete)
  (if (get-buffer mh-show-buffer)
      (kill-buffer mh-show-buffer))
  (kill-buffer (current-buffer)))
```

Kill MH-Folder buffer instead of burying it

You can use dired to manipulate the folders themselves. For example, I renamed my '`+out`' folder to the more common '`+outbox`' by running dired on my mail directory (*M-x dired RET ~/Mail RET*), moving my cursor to '`out`' and using the command *R* (`dired-do-rename`).

---

[3] See the section Sorting Messages: sortm in the MH book.

8 Sending Mail

You can send a mail message in several ways. You can call `M-x mh-smail` directly, or from the command line like this:

```
$ emacs -f mh-smail
```

There are some commands that need to send a mail message, such as `goto-address-at-point`. You can configure Emacs to have these commands use MH-E by setting the option `mail-user-agent` to 'Emacs interface to MH'.

From within MH-E's MH-Folder mode, other methods of sending mail are available as well. These can also be found in the 'Message' menu.

e   Edit a message to send it again (`mh-edit-again`).

E   Edit a message that was returned by the mail system (`mh-extract-rejected-mail`).

f   Forward message (`mh-forward`).

r   Reply to a message (`mh-reply`).

s   Compose a message (`mh-send`).

M-d   Redistribute a message (`mh-redistribute`).

M-x mh-smail
              Compose a message with the MH mail system.

M-x mh-smail-other-window
              Compose a message with the MH mail system in other window.

In addition, several options from the 'mh-sending-mail' customization group are useful when sending mail or replying to mail. They are summarized in the following table.

`mh-compose-forward-as-mime-flag`
              On means that messages are forwarded as attachments (default: 'on').

`mh-compose-letter-function`
              Hook run when starting a new draft (default: `nil`).

`mh-compose-prompt-flag`
              On means prompt for header fields when composing a new draft (default: 'off').

`mh-forward-subject-format`
              Format string for forwarded message subject (default: `"%s: %s"`).

`mh-insert-x-mailer-flag`
              On means append an 'X-Mailer:' header field to the header (default: 'on').

`mh-redist-full-contents-flag`
              On means the `dist` command needs entire letter for redistribution (default: 'off').

`mh-reply-default-reply-to`
              Sets the person or persons to whom a reply will be sent (default: 'Prompt').

`mh-reply-show-message-flag`
> On means the MH-Show buffer is displayed using `r` (`mh-reply`) (default: 'on').

The following hooks are available.

`mh-forward-hook`
> Hook run by `mh-forward` on a forwarded letter (default: `nil`).

`mh-letter-mode-hook`
> Hook run by `mh-letter-mode` on a new letter (default: `nil`).

The functions and options introduced here are explained in more detail in the following sections.

8.1 Composing

Outside of an MH-Folder buffer, you must call either *M-x mh-smail* or *M-x mh-smail-other-window* to compose a new message. The former command always creates a two-window layout with the current buffer on top and the draft on the bottom. Use the latter command if you would rather preserve the window layout. You may find adding the following key bindings to '`~/.emacs`' useful:

```
(global-set-key "\C-xm" 'mh-smail)
(global-set-key "\C-x4m" 'mh-smail-other-window)
```

From within a MH-Folder buffer, you can simply use the command *m* (`mh-send`). However you invoke `mh-send`, your letter appears in an Emacs buffer whose mode is MH-Letter (to see what the buffer looks like, see Section 3.1 [Sending Mail Tour], page 7). MH-Letter mode allows you to edit your message, to check the validity of the recipients, to insert attachments and other messages into your message, and to send the message. We'll go more into depth about editing a *draft*[1] (a message you're composing) in just a moment (see Chapter 9 [Editing Drafts], page 45).

If you prefer to be prompted for the recipient and subject fields before the MH-Letter buffer appears, turn on the option `mh-compose-prompt-flag`.

MH-E adds an 'X-Mailer:' header field to the header that includes the version of MH-E and Emacs that you are using. If you don't want to participate in our marketing, you can turn off the option `mh-insert-x-mailer-flag`.

Two hooks are provided to run commands on your freshly created draft. The first hook, `mh-letter-mode-hook`, allows you to do some processing before editing a letter[2]. For example, you may wish to modify the header after `repl` has done its work, or you may have a complicated '`components`' file and need to tell MH-E where the cursor should go. Here's an example of how you would use this hook.

---

[1] I highly recommend that you use a *draft folder* so that you can edit several drafts in parallel. To do so, create a folder named '`+drafts`' for example, and add the profile component '`Draft-Folder: drafts`' (see `mh-profile`(5)).

[2] Actually, because MH-Letter mode inherits from Mail mode, the hooks `text-mode-hook` and `mail-mode-hook` are run (in that order) before `mh-letter-mode-hook`.

```
(defvar letter-mode-init-done-flag nil
  "Non-nil means one-time MH-E settings have been made.")

(defun my-mh-letter-mode-hook ()
  "Prepare letter for editing."
  (when (not letter-mode-init-done)     ; only need to bind the keys once
    (local-set-key "\C-ctb" 'add-enriched-text)
    (local-set-key "\C-cti" 'add-enriched-text)
    (local-set-key "\C-ctf" 'add-enriched-text)
    (local-set-key "\C-cts" 'add-enriched-text)
    (local-set-key "\C-ctB" 'add-enriched-text)
    (local-set-key "\C-ctu" 'add-enriched-text)
    (local-set-key "\C-ctc" 'add-enriched-text)
    (setq letter-mode-init-done t))
  (save-excursion
    (goto-char (point-max))             ; go to end of message to
    (mh-insert-signature)))             ;    insert signature
```

Prepare draft for editing via mh-letter-mode-hook

The function, `add-enriched-text` is defined in the example in Section 9.6 [Adding Attachments], page 52.

The second hook, a function really, is `mh-compose-letter-function`. Like `mh-letter-mode-hook`, it is called just before editing a new message; however, it is the last function called before you edit your message. The consequence of this is that you can write a function to write and send the message for you. This function is passed three arguments: the contents of the 'To:', 'Subject:', and 'Cc:' header fields.

8.2 Replying to Mail

To compose a reply to a message, use the *r* (`mh-reply`) command.

When you reply to a message, you are first prompted with 'Reply to whom?'. You have several choices here.

Response	Reply Goes To
from	The person who sent the message. This is the default, so (RET) is sufficient.
to	Replies to the sender, plus all recipients in the 'To:' header field.
cc	Forms a reply to the addresses in the 'Mail-Followup-To:' header field
all	if one exists; otherwise forms a reply to the sender, plus all recipients.

Depending on your answer, `repl`[3] is given a different argument to form your reply. Specifically, a choice of *from* or none at all runs 'repl -nocc all', and a choice of *to* runs 'repl -cc to'. Finally, either *cc* or *all* runs 'repl -cc all -nocc me'. If you find that most of the time you specify one of these choices when you reply to a message, you can change the option `mh-reply-default-reply-to` from its default value of 'Prompt' to one of the choices listed above. You can always edit the recipients in the draft.

Two windows are then created. One window contains the message to which you are replying in an MH-Show buffer. Your draft, in MH-Letter mode (see Chapter 9 [Editing

---

[3] See the section Replying to Messages: repl in the MH book.

Drafts], page 45), is in the other window. If the reply draft was not one that you expected, check the things that affect the behavior of `repl` which include the '`repl:`' profile component and the '`replcomps`' and '`replgroupcomps`' files.

If you supply a prefix argument (as in `C-u r`), the message you are replying to is inserted in your reply after having first been run through `mhl` with the format file '`mhl.reply`'. See `mhl`(1) or the section Using mhl in the MH book to see how you can modify the default '`mhl.reply`' file.

Alternatively, you can customize the option `mh-yank-behavior` and choose one of its '`Automatically`' variants to do the same thing. See Section 9.2 [Inserting Letter], page 49. If you do so, the prefix argument has no effect.

Another way to include the message automatically in your draft is to use '`repl: -filter repl.filter`' in your MH profile.

If you include the message automatically, you can hide the MH-Show buffer by turning off the option `mh-reply-show-message-flag`.

If you wish to customize the header or other parts of the reply draft, please see `repl`(1) and `mh-format`(5).

8.3 Forwarding Mail

To forward a message, use the `f` (`mh-forward`) command. You are prompted for the '`To:`' and '`cc:`' recipients. You are given a draft to edit that looks like it would if you had run the MH command `forw`[4]. You can then add some text (see Chapter 9 [Editing Drafts], page 45). You can forward several messages by using a range (see Section 4.2 [Ranges], page 12). All of the messages in the range are inserted into your draft. The hook `mh-forward-hook` is called on the draft.

By default, the option `mh-compose-forward-as-mime-flag` is on which means that the forwarded messages are included as attachments. If you would prefer to forward your messages verbatim (as text, inline), then turn off this option. Forwarding messages verbatim works well for short, textual messages, but your recipient won't be able to view any non-textual attachments that were in the forwarded message. Be aware that if you have '`forw: -mime`' in your MH profile, then forwarded messages will always be included as attachments regardless of the settings of `mh-compose-forward-as-mime-flag`.

The format of the '`Subject:`' header field for forwarded messages is controlled by the option `mh-forward-subject-format`. This option is a string which includes two escapes ('`%s`'). The first '`%s`' is replaced with the sender of the original message, and the second one is replaced with the original '`Subject:`'. The default value of `"%s: %s"` takes a message with the header:

```
To: Bill Wohler <wohler@stop.mail-abuse.org>
Subject: Re: 49er football
From: Greg DesBrisay <gd@stop.mail-abuse.org>
```

and creates a subject header field of:

```
Subject: Greg DesBrisay: Re: 49er football
```

---

[4] See the section Forwarding Messages: forw in the MH book.

8.4 Redistributing Your Mail

The command `M-d` (`mh-redistribute`) is similar in function to forwarding mail, but it does not allow you to edit the message, nor does it add your name to the 'From:' header field. It appears to the recipient as if the message had come from the original sender. When you run this command, you are prompted for the recipients.

For more information on redistributing messages, see `dist`(1). Also investigate the command `e` (`mh-edit-again`) for another way to redistribute messages (see Section 8.5 [Editing Again], page 43).

The option `mh-redist-full-contents-flag` must be turned on if `dist`[5] requires the whole letter for redistribution, which is the case if `send`[6] is compiled with the BERK option (which many people abhor). If you find that MH will not allow you to redistribute a message that has been redistributed before, turn off this option.

8.5 Editing Old Drafts and Bounced Messages

If you don't complete a draft for one reason or another, and if the draft buffer is no longer available, you can pick your draft up again with `e` (`mh-edit-again`). If you don't use a draft folder, your last 'draft' file will be used. If you use draft folders, you'll need to visit the draft folder with `F v drafts` (RET), use `n` to move to the appropriate message, and then use `e` to prepare the message for editing.

The `e` command can also be used to take messages that were sent to you and to send them to more people.

Don't use `e` to re-edit a message from a *Mailer-Daemon* who complained that your mail wasn't posted for some reason or another. In this case, use `E` (`mh-extract-rejected-mail`) to prepare the message for editing by removing the *Mailer-Daemon* envelope and unneeded header fields. Fix whatever addressing problem you had, and send the message again with `C-c C-c`.

---

[5] See the section Distributing Messages with dist in the MH book.
[6] See the section Sending Some Mail: comp send in the MH book.

9 Editing a Draft

When you edit a message that you want to send (called a *draft* in this case), the mode used is MH-Letter. This mode provides several commands in addition to the normal Emacs editing commands to help you edit your draft. These can also be found in the 'Letter' menu.

⟨SPC⟩   Perform completion or insert space (`mh-letter-complete-or-space`).

M-⟨TAB⟩   Perform completion on header field or word preceding point (`mh-letter-complete`).

, (comma)   Flash alias expansion (`mh-letter-confirm-address`).

⟨TAB⟩   Cycle to next field (`mh-letter-next-header-field-or-indent`).

S-⟨TAB⟩   Cycle to the previous header field (`mh-letter-previous-header-field`).

C-c ?   Display cheat sheet for the MH-E commands (`mh-help`).

C-c C-c   Save draft and send message (`mh-send-letter`).

C-c C-d   Insert fields specified by the given identity (`mh-insert-identity`). See Chapter 11 [Identities], page 61.

C-c C-e   Compose MIME message from MH-style directives (`mh-mh-to-mime`).

C-c C-f C-a
C-c C-f a   Move to 'Mail-Reply-To:' header field (`mh-to-field`).

C-c C-f C-b
C-c C-f b   Move to 'Bcc:' header field (`mh-to-field`).

C-c C-f C-c
C-c C-f c   Move to 'Cc:' header field (`mh-to-field`).

C-c C-f C-d
C-c C-f d   Move to 'Dcc:' header field (`mh-to-field`).

C-c C-f C-f
C-c C-f f   Move to 'Fcc:' header field (`mh-to-fcc`).

C-c C-f C-l
C-c C-f l   Move to 'Mail-Followup-To:' header field (`mh-to-field`).

C-c C-f C-m
C-c C-f m   Move to 'From:' header field (`mh-to-field`).

C-c C-f C-r
C-c C-f r   Move to 'Reply-To:' header field (`mh-to-field`).

C-c C-f C-s
C-c C-f s   Move to 'Subject:' header field (`mh-to-field`).

C-c C-f C-t
C-c C-f t   Move to 'To:' header field (`mh-to-field`).

C-c C-i   Insert a message (`mh-insert-letter`).

C-c C-m C-e
           Add tag to encrypt the message (`mh-mml-secure-message-encrypt`).

C-c C-m C-f
C-c C-m f   Add tag to forward a message (`mh-compose-forward`).

C-c C-m C-g
C-c C-m g   Add tag to include anonymous ftp reference to a file (`mh-mh-compose-anon-`
           `ftp`).

C-c C-m C-i
C-c C-m i   Add tag to include a file such as an image or sound (`mh-compose-insertion`).

C-c C-m C-m
C-c C-m m   Compose MIME message from MML tags (`mh-mml-to-mime`).

C-c C-m C-n
C-c C-m n   Remove any secure message tags (`mh-mml-unsecure-message`).

C-c C-m C-s
           Add tag to sign the message (`mh-mml-secure-message-sign`).

C-c C-m C-t
C-c C-m t   Add tag to include anonymous ftp reference to a compressed tar file (`mh-mh-`
           `compose-external-compressed-tar`).

C-c C-m C-u
C-c C-m u   Undo effects of *C-c C-e* (`mh-mh-to-mime-undo`).

C-c C-m C-x
C-c C-m x   Add tag to refer to a remote file (`mh-mh-compose-external-type`).

C-c C-m e e
           Add tag to encrypt the message (`mh-mml-secure-message-encrypt`).

C-c C-m e s
           Add tag to encrypt and sign the message
           (`mh-mml-secure-message-signencrypt`).

C-c C-m s e
           Add tag to encrypt and sign the message
           (`mh-mml-secure-message-signencrypt`).

C-c C-m s s
           Add tag to sign the message (`mh-mml-secure-message-sign`).

C-c C-o   Insert a newline and leave point before it (`mh-open-line`).

C-c C-q   Quit editing and delete draft message (`mh-fully-kill-draft`).

C-c C-s   Insert signature in message (`mh-insert-signature`).

C-c C-t   Toggle display of header field at point (`mh-letter-toggle-header-field-`
           `display`).

C-c C-w   Verify recipients, showing expansion of any aliases (`mh-check-whom`).

C-c C-y   Insert the current message into the draft buffer (`mh-yank-cur-msg`).

C-c M-d   Insert custom fields if recipient is found in `mh-auto-fields-list` (`mh-insert-auto-fields`). See Chapter 11 [Identities], page 61.

   Several options from the '`mh-letter`' customization group are used while editing a draft.

`mh-compose-insertion`
           Type of MIME message tags in messages (default: 'MML' if available; otherwise
           'MH').

`mh-compose-skipped-header-fields`
           List of header fields to skip over when navigating in draft (default: '("From"
           "Organization" "References" "In-Reply-To" "X-Face" "Face" "X-Image-
           URL" "X-Mailer").

`mh-compose-space-does-completion-flag`
           On means (SPC) does completion in message header (default: 'off').

`mh-delete-yanked-msg-window-flag`
           On means delete any window displaying the message (default: 'off').

`mh-extract-from-attribution-verb`
           Verb to use for attribution when a message is yanked by *C-c C-y* (default:
           "wrote:").

`mh-ins-buf-prefix`
           String to put before each line of a yanked or inserted message (default: "> ").

`mh-letter-complete-function`
           Function to call when completing outside of address or folder fields (default:
           `ispell-complete-word`).

`mh-letter-fill-column`
           Fill column to use in MH-Letter mode (default: 72).

`mh-mml-method-default`
           Default method to use in security tags (default: 'PGP (MIME)' if support for it
           is available; otherwise 'None').

`mh-signature-file-name`
           Source of user's signature (default: "~/.signature").

`mh-signature-separator-flag`
           On means a signature separator should be inserted (default: 'on').

`mh-x-face-file`
           File containing X-Face or Face header field to insert in outgoing mail. (default:
           "~/.face").

`mh-yank-behavior`
           Controls which part of a message is yanked by *C-c C-y* (default: 'Body With
           Attribution').

   The following hooks are available.

`mail-citation-hook`
>   Hook for modifying a citation just inserted in the mail buffer (default: `nil`).

`mh-before-send-letter-hook`
>   Hook run at the beginning of the `C-c C-c` command (default: '`nil`').

`mh-mh-to-mime-hook`
>   Hook run on the formatted letter by `C-c C-e` (default: '`nil`').

`mh-insert-signature-hook`
>   Hook run by `C-c C-s` after signature has been inserted (default: `nil`).

The following face is available.

`mh-letter-header-field`
>   Editable header field value face in draft buffers.

The commands and options introduced here are explained in more detail in the following sections.

9.1 Editing the Message

Because the header is part of the message, you can edit the header fields as you wish. However, several convenience commands exist to help you create and edit them. For example, the command `C-c C-f C-t` (`mh-to-field`; alternatively, `C-c C-f t`) moves the cursor to the '`To:`' header field, creating it if necessary. The commands for moving to the '`Cc:`', '`Subject:`', '`From:`', '`Reply-To:`', '`Mail-Reply-To:`', '`Mail-Followup-To`', '`Bcc:`', and '`Dcc:`' header fields are similar.

One command behaves differently from the others, namely, `C-c C-f C-f` (`mh-to-fcc`; alternatively, `C-c C-f f`). This command will prompt you for the folder name in which to file a copy of the draft. See Section 4.3 [Folder Selection], page 13.

Within the header of the message, the command ⟨TAB⟩ (`mh-letter-next-header-field-or-indent`) moves between fields that are highlighted with the face `mh-letter-header-field`, skipping those fields listed in `mh-compose-skipped-header-fields`. After the last field, this command then moves point to the message body before cycling back to the first field. If point is already past the first line of the message body, then this command indents by calling `indent-relative` with the given prefix argument. The command `S-`⟨TAB⟩ (`mh-letter-previous-header-field`) moves backwards between the fields and cycles to the body of the message after the first field. Unlike the command ⟨TAB⟩, it will always take point to the last field from anywhere in the body.

If the field contains addresses (for example, '`To:`' or '`Cc:`') or folders (for example, '`Fcc:`') then the command `M-`⟨TAB⟩ (`mh-letter-complete`) will provide alias completion (see Chapter 10 [Aliases], page 57). In the body of the message, `M-`⟨TAB⟩ runs `mh-letter-complete-function` instead, which is set to ''`ispell-complete-word`' by default. The command `M-`⟨TAB⟩ (`mh-letter-complete`) takes a prefix argument that is passed to the `mh-letter-complete-function`. In addition, turn on the option `mh-compose-space-does-completion-flag` to use the command ⟨SPC⟩ (`mh-letter-complete-or-space`) to perform completion in the header as well; use a prefix argument to specify more than one space. Addresses are separated by a comma; when you press the comma, the command `mh-letter-confirm-address` flashes the alias expansion in the minibuffer if `mh-alias-flash-on-comma` is turned on.

Use the command `C-c C-t` `mh-letter-toggle-header-field-display` to display trun-
cated header fields. This command is a toggle so entering it again will hide the field. This
command takes a prefix argument: if negative then the field is hidden, if positive then the
field is displayed (for example, `C-u C-c C-t`).

Be sure to leave a row of dashes or a blank line between the header and the body of the
message.

The body of the message is edited as you would edit any Emacs buffer although there
are a few commands and options to assist you. You can change the fill column in MH-Letter
mode with the option `mh-letter-fill-column`. By default, this option is 72 to allow others
to quote your message without line wrapping.

You'll often include messages that were sent from user agents that haven't yet realized
that paragraphs consist of more than a single line. This makes for long lines that wrap in
an ugly fashion. You'll find that `M-q` (`fill-paragraph`) works well even on these quoted
messages, even if they are nested, just as long as all of the quotes match the value of `mh-`
`ins-buf-prefix` (see Section 9.2 [Inserting Letter], page 49). For example, let's assume
you have the following in your draft:

```
> Hopefully this gives you an idea of what I'm currently doing. I'm \
not sure yet whether I'm completely satisfied with my setup, but     \
it's worked okay for me so far.
```

Running `M-q` on this paragraph produces:

```
> Hopefully this gives you an idea of what I'm currently doing. I'm not
> sure yet whether I'm completely satisfied with my setup, but it's
> worked okay for me so far.
```

The command `C-c C-o` (`mh-open-line`) is similar to the command `C-o` (`open-line`) in
that it inserts a newline after point. It differs in that it also inserts the right number of
quoting characters and spaces so that the next line begins in the same column as it was.
This is useful when breaking up paragraphs in replies. For example, if this command was
used when point was after the first period in the paragraph above, the result would be this:

```
> Hopefully this gives you an idea of what I'm currently doing.

>                                                     I'm not
> sure yet whether I'm completely satisfied with my setup, but it's
> worked okay for me so far.
```

9.2 Inserting Letter to Which You're Replying

It is often useful to insert a snippet of text from a letter that someone mailed to provide
some context for your reply. The command `C-c C-y` (`mh-yank-cur-msg`) does this by adding
an attribution, yanking a portion of text from the message to which you're replying, and
inserting `mh-ins-buf-prefix` ('> ') before each line.

```
Michael W Thelen <thelenm@stop.mail-abuse.org> wrote:

> Hopefully this gives you an idea of what I'm currently doing. I'm not
> sure yet whether I'm completely satisfied with my setup, but it's
> worked okay for me so far.
```

The attribution consists of the sender's name and email address followed by the content
of the option `mh-extract-from-attribution-verb`. This option can be set to '`wrote:`',
'`a crit:`', and '`schrieb:`'. You can also use the '`Custom String`' menu item to enter your
own verb.

The prefix `"> "` is the default setting for the option `mh-ins-buf-prefix`. I suggest that you not modify this option since it is used by many mailers and news readers: messages are far easier to read if several included messages have all been indented by the same string. This prefix is not inserted if you use one of the supercite flavors of `mh-yank-behavior` or you have added a `mail-citation-hook` as described below.

You can also turn on the `mh-delete-yanked-msg-window-flag` option to delete the window containing the original message after yanking it to make more room on your screen for your reply.

You can control how the message to which you are replying is yanked into your reply using `mh-yank-behavior`. To include the entire message, including the entire header, use 'Body and Header'[1][2]. Use 'Body' to yank just the body without the header. To yank only the portion of the message following the point, set this option to 'Below Point'.

Choose 'Invoke supercite'[3] to pass the entire message and header through supercite.

If the 'Body With Attribution' setting is used, then the message minus the header is yanked and a simple attribution line is added at the top using the value of the option `mh-extract-from-attribution-verb`. This is the default.

If the 'Invoke supercite' or 'Body With Attribution' settings are used, the '-noformat' argument is passed to the `repl` program to override a '-filter' or '-format' argument. These settings also have 'Automatically' variants that perform the action automatically when you reply so that you don't need to use `C-c C-y` at all. Note that this automatic action is only performed if the show buffer matches the message being replied to. People who use the automatic variants tend to turn on the option `mh-delete-yanked-msg-window-flag` as well so that the show window is never displayed.

If the show buffer has a region, the option `mh-yank-behavior` is ignored unless its value is one of 'Attribution' variants in which case the attribution is added to the yanked region.

If this isn't enough, you can gain full control over the appearance of the included text by setting `mail-citation-hook` to a function that modifies it. This hook is ignored if the option `mh-yank-behavior` is set to one of the supercite flavors. Otherwise, this option controls how much of the message is passed to the hook. The function can find the citation between point and mark and it should leave point and mark around the modified citation text for the next hook function. The standard prefix `mh-ins-buf-prefix` is not added if this hook is set.

For example, if you use the hook function `trivial-cite` (which is NOT part of Emacs), set `mh-yank-behavior` to 'Body and Header'.

9.3 Inserting Messages

Messages can be inserted with `C-c C-i` (`mh-insert-letter`). This command prompts you for the folder and message number, which defaults to the current message in that folder. It then inserts the messages, indented by `mh-ins-buf-prefix` ('> ') unless `mh-yank-behavior`

---

[1] If you'd rather have the header cleaned up, use `C-u r` instead of `r` when replying (see Section 8.2 [Replying], page 41).

[2] In the past you would use this setting and set `mail-citation-hook` to 'supercite', but this usage is now deprecated in favor of the 'Invoke supercite' setting.

[3] *Supercite* is a full-bodied, full-featured, citation package that comes standard with Emacs.

is set to one of the supercite flavors in which case supercite is used to format the message. Certain undesirable header fields (see `mh-invisible-header-fields-compiled`) are removed before insertion.

If given a prefix argument (like `C-u C-c C-i`), the header is left intact, the message is not indented, and '`>` ' is not inserted before each line. This command leaves the mark before the letter and point after it.

9.4 Inserting Your Signature

You can insert your signature at the current cursor location with the command `C-c C-s` (`mh-insert-signature`).

By default, the text of your signature is taken from the file '`~/.signature`'. You can read from other sources by changing the option `mh-signature-file-name`. This file may contain a *vCard* in which case an attachment is added with the vCard.

The option `mh-signature-file-name` may also be a symbol, in which case that function is called. You may not want a signature separator to be added for you; instead you may want to insert one yourself. Options that you may find useful to do this include `mh-signature-separator` (when inserting a signature separator) and `mh-signature-separator-regexp` (for finding said separator). The function `mh-signature-separator-p`, which reports `t` if the buffer contains a separator, may be useful as well.

A signature separator (`"-- "`) will be added if the signature block does not contain one and `mh-signature-separator-flag` is on. It is not recommended that you change this option since various mail user agents, including MH-E, use the separator to present the signature differently, and to suppress the signature when replying or yanking a letter into a draft.

The hook `mh-insert-signature-hook` is run after the signature is inserted. Hook functions may access the actual name of the file or the function used to insert the signature with `mh-signature-file-name`.

The signature can also be inserted using Identities. See Chapter 11 [Identities], page 61.

9.5 Inserting Your Picture

You can insert your picture in the header of your mail message so that recipients see your face in the '`From:`' header field if their mail user agent is sophisticated enough. In MH-E, this is done by placing your image in the file named by the option `mh-x-face-file` which is '`~/.face`' by default.

If the file starts with either of the strings '`X-Face:`', '`Face:`' or '`X-Image-URL:`' then the contents are added to the message header verbatim. Otherwise it is assumed that the file contains the value of the '`X-Face:`' header field.

The '`X-Face:`' header field, which is a low-resolution, black and white image, can be generated using the `compface` command. The *Online X-Face Converter* is a useful resource for quick conversion of images into '`X-Face:`' header fields.

Use the `make-face` script to convert a JPEG image to the higher resolution, color, '`Face:`' header field.

The URL of any image can be used for the '`X-Image-URL:`' field and no processing of the image is required.

To prevent the setting of any of these header fields, either set `mh-x-face-file` to `nil`, or simply ensure that the file defined by this option doesn't exist.

See Section 6.1 [Viewing], page 21, to see how these header fields are displayed in MH-E.

9.6 Adding Attachments

MH-E has the capability to create multimedia messages. It uses the MIME (Multipurpose Internet Mail Extensions) protocol[4] The MIME protocol allows you to incorporate images, sound, video, binary files, and even commands that fetch a file with '`ftp`' when your recipient reads the message!

If you were to create a multimedia message with plain MH commands, you would insert `mhbuild` or `mhn` directives (henceforth called *MH-style directives* into your draft and use the `mhbuild` command in nmh or `mhn` command in MH and GNU mailutils to expand them. MH-E works in much the same way, although it provides a handful of commands prefixed with `C-c C-m` to insert the directives so you don't need to remember the syntax of them. Remember: you can always add MH-style directives by hand[5].

In addition to MH-style directives, MH-E also supports MML (MIME Meta Language) tags[6]. The option `mh-compose-insertion` can be used to choose between them. By default, this option is set to 'MML' if it is supported since it provides a lot more functionality. This option can also be set to 'MH' if MH-style directives are preferred.

The MH-E MIME commands require a *media type* for each body part or attachment. For example, a PDF document is of type '`application/pdf`' and an HTML document is of type '`text/html`'. Some commands fill in the media type for you, whereas others require you to enter one.

In the cases where MH-E can do so, it will determine the media type automatically. It uses the `file` command to do this. Failing that, the Emacs function `mailcap-mime-types` is used to provide a list from which to choose. This function usually reads the file '`/etc/mime.types`'.

Whether the media type is chosen automatically, or you choose it from a list, use the type that seems to match best the file that you are including. In the case of binaries, the media type '`application/x-executable`' can be useful. If you can't find an appropriate media type, use '`text/plain`' for text messages and '`application/octet-stream`' for everything else.

You are also sometimes asked for a *content description*. This is simply an optional brief phrase, in your own words, that describes the object. If you don't care to enter a content description, just press return and none will be included; however, a reader may skip over multimedia fields unless the content description is compelling.

You can also create your own MIME body parts. In the following example, I describe how you can create and edit a '`text/enriched`' body part to liven up your plain text messages with boldface, underlining, and italics. I include an Emacs function which inserts enriched text tags.

---

[4] MIME is defined in RFC 2045.

[5] See the section Sending MIME Mail in the MH book.

[6]   See the section Composing in *The Emacs MIME Manual*.

```
(defvar enriched-text-types '(("b" . "bold") ("i" . "italic")
                              ("u" . "underline")
                              ("s" . "smaller") ("B" . "bigger")
                              ("f" . "fixed")
                              ("c" . "center"))
  "Alist of (final-character . tag) choices for add-enriched-text.
Additional types can be found in RFC 1563.")

(defun add-enriched-text (begin end)
  "Add enriched text tags around region.
The tag used comes from the list enriched-text-types and is
specified by the last keystroke of the command.  When called from Lisp,
arguments are BEGIN and END."
  (interactive "r")
  ;; Set type to the tag indicated by the last keystroke.
  (let ((type (cdr (assoc (char-to-string (logior last-input-char ?'))
                          enriched-text-types))))
    (save-restriction                   ; restores state from narrow-to-region
      (narrow-to-region begin end)      ; narrow view to region
      (goto-char (point-min))           ; move to beginning of text
      (insert "<" type ">")             ; insert beginning tag
      (goto-char (point-max))           ; move to end of text
      (insert "</" type ">"))))         ; insert terminating tag
Emacs function for entering enriched text
```

To use the function `add-enriched-text`, first add it to '`~/.emacs`' and create key bindings for it (see Section 8.1 [Composing], page 40).

Then, in your plain text message, set the mark with `C-@` or `C-`$\overline{\text{SPC}}$, type in the text to be highlighted, and type `C-c t b`. This adds '`<bold>`' where you set the mark and adds '`</bold>`' at the location of your cursor, giving you something like: '`You should be <bold>very</bold>`'.

Before sending this message, use `C-c C-m C-m` (`mh-mml-to-mime`)[7] to add MIME header fields. Then replace '`text/plain`' with '`text/enriched`' in the '`Content-Type:`' header field.

You may also be interested in investigating `sgml-mode`.

Including Files

Binaries, images, sound, and video can be inserted in your message with the command `C-c C-m C-i` (`mh-compose-insertion`). You are prompted for the filename containing the object, the media type if it cannot be determined automatically, and a content description. If you're using MH-style directives, you will also be prompted for additional attributes.

Forwarding Multimedia Messages

Mail may be forwarded with MIME using the command `C-c C-m C-f` (`mh-compose-forward`). You are prompted for a content description, the name of the folder in which the messages to forward are located, and a range of messages, which defaults to the current message in that folder. See Section 4.2 [Ranges], page 12.

---

[7] Use `C-c C-e` (`mh-mh-to-mime`) if you're using MH-style directives.

Including an FTP Reference

You can have your message initiate an `ftp` transfer when the recipient reads the message. To do this, use the command *C-c C-m C-g* (`mh-mh-compose-anon-ftp`). You are prompted for the remote host and filename, the media type, and the content description.

Including tar Files

If the remote file is a compressed tar file, you can use *C-c C-m C-t* (`mh-mh-compose-external-compressed-tar`). Then, in addition to retrieving the file via anonymous *ftp* as per the command *C-c C-m C-g* (`mh-mh-compose-anon-ftp`), the file will also be uncompressed and untarred. You are prompted for the remote host and filename and the content description.

Including Other External Files

The command *C-c C-m C-x* (`mh-mh-compose-external-type`) is a general utility for referencing external files. In fact, all of the other commands that insert tags to access external files call this command. You are prompted for the access type, remote host and filename, and content type. If you provide a prefix argument, you are also prompted for a content description, attributes, parameters, and a comment.

Previewing Multimedia Messages

When you are finished editing a MIME message, it might look like this:

```
3 t08/24   root                  received fax files on Wed Aug 24 11:00:
4+t08/24  To:wohler              Test<<This is a test message to get the




--:%%  {+inbox} 4 msgs (1-4)   Bot L4      (MH-Folder Show)---------------
To: wohler
cc:
Subject: Test of MIME
--------
Here is the SETI@Home logo:

<#part type="image/x-xpm" filename="~/lib/images/setiathome.xpm"
disposition=inline description="SETI@home logo">
<#/part>
--:**  {draft}   All L8      (MH-Letter)---------------------------------
```

MH-E MIME *draft*

Typically, you send a message with attachments just like any other message (see Section 9.9 [Sending Message], page 56).

However, you may take a sneak preview of the MIME encoding if you wish by running the command *C-c C-m C-m* (`mh-mml-to-mime`). The following screen shows the MIME encoding specified by the tags. You can see why mail user agents are usually built to hide these details from the user.

```
To: wohler
cc:
Subject: Test of MIME
X-Mailer: MH-E 8.0; nmh 1.1; GNU Emacs 22.1
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="=-=-="
--------
--=-=-=

Here is the SETI@Home logo:


--=-=-=
Content-Type: image/x-xpm
Content-Disposition: inline; filename=setiathome.xpm
Content-Transfer-Encoding: base64
Content-Description: SETI@home logo

LyogWFBNICovCnN0YXRpYyBjaGFyICogc2V0aWF0aG9tZV94cG1bXSA9IHsKIjQ1IDQ1IDc2N
--:--  {draft}   Top L1     (MH-Letter)---------------------------------
```

MH-E MIME *draft ready to send*

This action can be undone by running `C-_` (`undo`).

If you're using MH-style directives, use `C-c C-e` (`mh-mh-to-mime`) instead of `C-c C-m C-m`. This runs the command `mhbuild` (`mhn`) on the message which expands the tags[8]. This action can be undone by running `C-c C-m C-u` (`mh-mh-to-mime-undo`), which works by reverting to a backup file. You are prompted to confirm this action, but you can avoid the confirmation by adding an argument (for example, `C-u C-c C-m C-u`).

If you wish to pass additional arguments to `mhbuild` (`mhn`) to affect how it builds your message, use the option `mh-mh-to-mime-args`. For example, you can build a consistency check into the message by setting `mh-mh-to-mime-args` to '`-check`'. The recipient of your message can then run '`mhbuild -check`' on the message—`mhbuild` (`mhn`) will complain if the message has been corrupted on the way. The command `C-c C-e` only consults this option when given a prefix argument (as in `C-u C-c C-e`).

The hook `mh-mh-to-mime-hook` is called after the message has been formatted by `C-c C-e`.

9.7 Signing and Encrypting Messages

MH-E can sign and encrypt messages as defined in RFC 3156. If you should choose to sign or encrypt your message, use one of the following commands to do so any time before sending your message.

The command `C-c C-m C-s` (`mh-mml-secure-message-sign`) inserts the following tag:

    <#secure method=pgpmime mode=sign>

This is used to sign your message digitally. Likewise, the command `C-c C-m C-e` (`mh-mml-secure-message-encrypt`) inserts the following tag:

    <#secure method=pgpmime mode=encrypt>

This is used to encrypt your message. Finally, the command `C-c C-m s e` (`mh-mml-secure-message-signencrypt`) inserts the following tag:

---

[8] See the section Sending MIME Mail in the MH book.

```
<#secure method=pgpmime mode=signencrypt>
```

This is used to sign and encrypt your message. In each of these cases, a proper multipart message is created for you when you send the message. Use the command `C-c C-m C-n` (`mh-mml-unsecure-message`) to remove these tags. Use a prefix argument (as in `C-u C-c C-m s e`) to be prompted for one of the possible security methods (see `mh-mml-method-default`).

The option `mh-mml-method-default` is used to select between a variety of mail security mechanisms. The default is 'PGP (MIME)' if it is supported; otherwise, the default is 'None'. Other mechanisms include vanilla 'PGP' and 'S/MIME'.

The 'pgg' customization group may have some settings which may interest you. See *The PGG Manual*.

In particular, I turn on the option `pgg-encrypt-for-me` so that all messages I encrypt are encrypted with my public key as well. If you keep a copy of all of your outgoing mail with a 'Fcc:' header field, this setting is vital so that you can read the mail you write!

9.8 Checking Recipients

The command `C-c C-w` (`mh-check-whom`) expands aliases so you can check the actual address(es) in the alias. A new buffer named '*MH-E Recipients*' is created with the output of `whom` (see Chapter 20 [Miscellaneous], page 89)[9].

9.9 Sending a Message

When you are all through editing a message, you send it with the command `C-c C-c` (`mh-send-letter`). You can give a prefix argument (as in `C-u C-c C-c`) to monitor the first stage of the delivery; this output can be found in a buffer called '*MH-E Mail Delivery*' (see Chapter 20 [Miscellaneous], page 89).

The hook `mh-before-send-letter-hook` is run at the beginning of the command `C-c C-c`. For example, if you want to check your spelling in your message before sending, add the function `ispell-message`.

In case the MH `send` program[10] is installed under a different name, use `mh-send-prog` to tell MH-E the name.

9.10 Killing the Draft

If for some reason you are not happy with the draft, you can use the command `C-c C-q` (`mh-fully-kill-draft`) to kill the draft buffer and delete the draft message. Use the command `C-x k` (`kill-buffer`) if you don't want to delete the draft message.

---

[9]　See the section What now? – and the whatnow Program in the MH book.

[10]　See the section Sending Some Mail: comp send in the MH book.

10 Aliases

MH aliases are used in the same way in MH-E as they are in MH. Any alias listed as a recipient will be expanded when the message is sent. This chapter discusses other things you can do with aliases in MH-E.

The following commands are available in MH-Letter mode with the exception of `mh-alias-reload` which can be called from anywhere.

⟨SPC⟩   Perform completion or insert space (`mh-letter-complete-or-space`).

M-⟨TAB⟩   Perform completion on header field or word preceding point (`mh-letter-complete`).

mh-alias-apropos
> Show all aliases or addresses that match a regular expression.

mh-alias-grab-from-field
> Add alias for the sender of the current message

mh-alias-reload
> Reload MH aliases.

The '`mh-alias`' customization group contains options associated with aliases.

`mh-alias-completion-ignore-case-flag`
> On means don't consider case significant in MH alias completion (default: '`on`').

`mh-alias-expand-aliases-flag`
> On means to expand aliases entered in the minibuffer (default: '`off`').

`mh-alias-flash-on-comma`
> Specify whether to flash address or warn on translation (default: '`Flash but Don't Warn If No Alias`').

`mh-alias-insert-file`
> Filename used to store a new MH-E alias (default: '`Use Aliasfile Profile Component`').

`mh-alias-insertion-location`
> Specifies where new aliases are entered in alias files (default: '`Alphabetical`').

`mh-alias-local-users`
> If '`on`', local users are added to alias completion (default: '`on`').

`mh-alias-local-users-prefix`
> String prefixed to the real names of users from the password file (default: `"local."`.

`mh-alias-passwd-gecos-comma-separator-flag`
> On means the GECOS field in the password file uses a comma separator (default: '`on`').

The following hook is available.

`mh-alias-reloaded-hook`
> Hook run by `mh-alias-reload` after loading aliases (default: `nil`).

Adding Addresses to Draft

You can use aliases when you are adding recipients to a message.

In order to use minibuffer prompting for recipients and the subject line in the minibuffer, turn on the option `mh-compose-prompt-flag` (see Section 8.1 [Composing], page 40), and use the ⟨TAB⟩ (`minibuffer-complete`) command to complete aliases (and optionally local logins) when prompted for the recipients. Turn on the option `mh-alias-expand-aliases-flag` if you want these aliases to be expanded to their respective addresses in the draft.

Otherwise, you can complete aliases in the header of the draft with M-⟨TAB⟩ (`mh-letter-complete`) or ⟨SPC⟩ (`mh-letter-complete-or-space`).

As MH ignores case in the aliases, so too does MH-E. However, you may turn off the option `mh-alias-completion-ignore-case-flag` to make case significant which can be used to segregate completion of your aliases. You might use uppercase for mailing lists and lowercase for people. For example, you might have:

```
mark.baushke: Mark Baushke <mdb@stop.mail-abuse.org>
MH-E: MH-E Mailing List <mh-e-devel@stop.mail-abuse.org>
```

When this option is turned off, if you were to type M in the 'To:' field and then M-⟨TAB⟩, then you'd get the list; if you started with m and then entered M-⟨TAB⟩, then you'd get Mark's address. Note that this option affects completion only. If you were to enter *Mark.Baushke*, it would still be identified with your '`mark.baushke`' alias.

To verify that the alias you've entered is valid, the alias will be displayed in the minibuffer when you type a comma (`mh-letter-confirm-address` or `mh-alias-minibuffer-confirm-address` if the option `mh-compose-prompt-flag` is turned on). See Section 8.1 [Composing], page 40. This behavior can be controlled with the option `mh-alias-flash-on-comma` which provides three choices: '`Flash but Don't Warn If No Alias`', '`Flash and Warn If No Alias`', and '`Don't Flash Nor Warn If No Alias`'.

For another way to verify the alias expansion, see Section 9.8 [Checking Recipients], page 56.

Loading Aliases

MH-E loads aliases for completion and folder name hints from various places. It uses the MH command `ali`[1] to read aliases from the files listed in the profile component '`Aliasfile:`' as well as system-wide aliases (for example, '`/etc/nmh/MailAliases`').

In addition, aliases are created from '`/etc/passwd`' entries with a user ID larger than a magical number, typically 200. This can be a handy tool on a machine where you and co-workers exchange messages. These aliases have the form '`local.first.last`' if a real name is present in the password file. Otherwise, the alias will have the form '`local.login`'.

The prefix '`local.`' can be modified via the option `mh-alias-local-users-prefix`. This option can also be set to '`Use Login`'.

For example, consider the following password file entry:

```
psg:x:1000:1000:Peter S Galbraith,,,:/home/psg:/bin/tcsh
```

The following settings of option `mh-alias-local-users-prefix` will produce the associated aliases:

---

[1] See the section MH Aliases in the MH book.

`"local."`   local.peter.galbraith

`""`   peter.galbraith

`Use Login` psg

In the example above, commas are used to separate different values within the so-called GECOS field. This is a fairly common usage. However, in the rare case that the GECOS field in your password file is not separated by commas and whose contents may contain commas, you can turn the option `mh-alias-passwd-gecos-comma-separator-flag` off.

If you're on a system with thousands of users you don't know, and the loading of local aliases slows MH-E down noticeably, then the local alias feature can be disabled by turning off the option `mh-alias-local-users`. This option also takes a string which is executed to generate the password file. For example, use '`ypcat passwd`' to obtain the NIS password file.

Since aliases are updated frequently, MH-E reloads aliases automatically whenever an alias lookup occurs if an alias source has changed. However, you can reload your aliases manually by calling the command *M-x mh-alias-reload* directly. This command runs `mh-alias-reloaded-hook` after the aliases have been loaded.

Adding Aliases

In the past, you have manually added aliases to your alias file(s) listed in your '`Aliasfile:`' profile component. MH-E provides other methods for maintaining your alias file(s).

You can use the *M-x mh-alias-add-alias* command which will prompt you for the alias and address that you would like to add. If the alias exists already, you will have the choice of inserting the new alias before or after the old alias. In the former case, this alias will be used when sending mail to this alias. In the latter case, the alias serves as an additional folder name hint when filing messages (see Section 4.3 [Folder Selection], page 13).

Earlier, the alias prefix '`local`' was presented. You can use other prefixes to organize your aliases or disambiguate entries. You might use prefixes for locales, jobs, or activities. For example, I have:

```
; Work
attensity.don.mitchell: Don Mitchell <dmitchell@stop.mail-abuse.com>
isharp.don.mitchell: Don Mitchell <donaldsmitchell@stop.mail-abuse.com>
...
; Sport
diving.ken.mayer: Ken Mayer <kmayer@stop.mail-abuse.com>
sailing.mike.maloney: Mike Maloney <mmaloney@stop.mail-abuse.com>
...
; Personal
ariane.kolkmann: Ariane Kolkmann <ArianeKolkmann@stop.mail-abuse.com>
...
```

Using prefixes instead of postfixes helps you explore aliases during completion. If you forget the name of an old dive buddy, you can enter '`div`' and then (SPC) to get a listing of all your dive buddies.

An alias for the sender of the current message is added automatically by clicking on the '`Grab From alias`' tool bar button or by running the *M-x mh-alias-grab-from-field* command. Aliases for other recipients of the current message are added by placing your

cursor over the desired recipient and giving the `M-x mh-alias-add-address-under-point` command.

The options `mh-alias-insert-file` and `mh-alias-insertion-location` controls how and where these aliases are inserted.

The default setting of option `mh-alias-insert-file` is 'Use Aliasfile Profile Component'. This option can also hold the name of a file or a list a file names. If this option is set to a list of file names, or the 'Aliasfile:' profile component contains more than one file name, MH-E will prompt for one of them.

The option `mh-alias-insertion-location` is set to 'Alphabetical' by default. If you organize your alias file in other ways, then the settings 'Top' and 'Bottom' might be more appropriate.

Querying Aliases

If you can't quite remember an alias, you can use `M-x mh-alias-apropos` to show all aliases or addresses that match a regular expression (see section "Syntax of Regular Expressions" in *The GNU Emacs Manual*).

11 Identities

MH-E supports the concept of multiple personalities or identities. This means that you can easily have a different header and signature at home and at work.

A couple of commands are used to insert identities in MH-Letter mode which are also found in the 'Identity' menu.

C-c C-d   Insert fields specified by given identity (`mh-insert-identity`).

C-c M-d   Insert custom fields if recipient found in `mh-auto-fields-list` (`mh-insert-auto-fields`).

The '`mh-identity`' customization group contains the following options.

`mh-auto-fields-list`
> List of recipients for which header lines are automatically inserted (default: `nil`).

`mh-auto-fields-prompt-flag`
> On means to prompt before sending if fields inserted (default: '`on`')

`mh-identity-default`
> Default identity to use when `mh-letter-mode` is called (default: '`None`').

`mh-identity-handlers`
> Handler functions for fields in `mh-identity-list`.

`mh-identity-list`
> List of identities (default: `nil`).

Some of the common header fields that people change depending on the context are the '`From:`' and '`Organization:`' fields, as well as the signature.

This is done by customizing the option `mh-identity-list`. In the customization buffer for this option, click on the '`INS`' button and enter a label such as '`Home`' or '`Work`'. Then click on the '`INS`' button with the label '`Add at least one item below`'. The '`Value Menu`' has the following menu items:

'`From Field`'
> Specify an alternate '`From:`' header field. You must include a valid email address. A standard format is '`First Last <login@host.domain>`'. If you use an initial with a period, then you must quote your name as in '`"First I. Last" <login@host.domain>`'.

'`Organization Field`'
> People usually list the name of the company where they work here.

'`Other Field`'
> Set any arbitrary header field and value here. Unless the header field is a standard one, precede the name of your field's label with '`X-`', as in '`X-Fruit-of-the-Day:`'.

'`Attribution Verb`'
> This value overrides the setting of `mh-extract-from-attribution-verb`. See Section 9.2 [Inserting Letter], page 49.

'`Signature`'
>   Set your signature with this item. You can specify the contents of
>   `mh-signature-file-name`, a file, or a function. See Section 9.4 [Signature],
>   page 51.

'`GPG Key ID`'
>   Specify a different key to sign or encrypt messages.

You can select the identities you have added via the menu called '`Identity`' in the MH-Letter buffer. You can also use *C-c C-d* (`mh-insert-identity`). To clear the fields and signature added by the identity, select the '`None`' identity.

The '`Identity`' menu contains two other items to save you from having to set the identity on every message. The menu item '`Set Default for Session`' can be used to set the default identity to the current identity until you exit Emacs. The menu item '`Save as Default`' sets the option `mh-identity-default` to the current identity setting. You can also customize the option `mh-identity-default` in the usual fashion. If you find that you need to add another identity, the menu item '`Customize Identities`' is available for your convenience.

The option `mh-auto-fields-list` can also be used to set the identity depending on the recipient to provide even more control. To customize `mh-auto-fields-list`, click on the '`INS`' button and enter a regular expression for the recipient's address (see section "Syntax of Regular Expressions" in *The GNU Emacs Manual*). Click on the '`INS`' button with the '`Add at least one item below`' label. The '`Value Menu`' contains the following menu items:

'`Identity`'
>   Select an identity from those configured in `mh-identity-list`. All of the information for that identity will be added if the recipient matches.

'`Fcc Field`'
>   Insert an '`Fcc:`' header field with the folder you provide. When you send the message, MH will put a copy of your message in this folder.

'`Mail-Followup-To Field`'
>   Insert an '`Mail-Followup-To:`' header field with the recipients you provide. If the recipient's mail user agent supports this header field[1], then their replies will go to the addresses listed. This is useful if their replies go both to the list and to you and you don't have a mechanism to suppress duplicates. If you reply to someone not on the list, you must either remove the '`Mail-Followup-To:`' field, or ensure the recipient is also listed there so that he receives replies to your reply.

'`Other Field`'
>   Other header fields may be added using this menu item.

These fields can only be added after the recipient is known. Because you can continue to add recipients as you edit the draft, MH-E waits until the message is sent to perform the auto-insertions. This seems strange at first, but you'll get used to it. There are two ways to help you feel that the desired fields are added. The first is the action when the

---

[1] '`Mail-Followup-To:`' is supported by nmh.

message is sent: if any fields are added automatically, you are given a chance to see and to confirm these fields before the message is actually sent. You can do away with this confirmation by turning off the option `mh-auto-fields-prompt-flag`. The second method is manual: once the header contains one or more recipients, you may run the command *C-c M-d* (`mh-insert-auto-fields`) or choose the 'Identity -> Insert Auto Fields' menu item to insert these fields manually. However, if you use this command, the automatic insertion when the message is sent is disabled.

You should avoid using the same header field in `mh-auto-fields-list` and `mh-identity-list` definitions that may apply to the same message as the result is undefined.

The option `mh-identity-handlers` is used to change the way that fields, signatures, and attributions in `mh-identity-list` are added. To customize `mh-identity-handlers`, replace the name of an existing handler function associated with the field you want to change with the name of a function you have written. You can also click on an 'INS' button and insert a field of your choice and the name of the function you have written to handle it.

The 'Field' field can be any field that you've used in your `mh-identity-list`. The special fields ':attribution-verb', ':signature', or ':pgg-default-user-id' are used for the `mh-identity-list` choices 'Attribution Verb', 'Signature', and 'GPG Key ID' respectively.

The handler associated with the ':default' field is used when no other field matches.

The handler functions are passed two or three arguments: the field itself (for example, 'From'), or one of the special fields (for example, ':signature'), and the action ''remove' or ''add'. If the action is ''add', an additional argument containing the value for the field is given.

12 The Speedbar

You can also use the speedbar (see section "Speedbar Frames" in *The GNU Emacs Manual*,) to view your folders. To bring up the speedbar, run *M-x speedbar* (RET). You will see a new frame appear with all of your MH folders. Folders with unseen messages appear in boldface. Click on a folder name with *Mouse-2* to visit that folder in a similar fashion to the command *F v* (`mh-visit-folder`) (see Chapter 7 [Folders], page 33). Click on the '+' icon to expand and view the sub-folders of that folder.

The speedbar can be manipulated with the keyboard as well. Use the Emacs navigational keys (like the arrow keys, or *C-n*) to move the cursor over the desired folder and then use the shortcuts for the menu items listed in the table below.

'Visit Folder ((RET))'
> Visits the selected folder just as if you had used *F v* (`mh-speed-view`).

'Expand Nested Folders (+)'
> Expands the selected folder in the speedbar, exposing the children folders inside it (`mh-speed-expand-folder`).

'Contract Nested Folders (-)'
> Contracts or collapses the selected folder in the speedbar, hiding the children folders inside it (`mh-speed-contract-folder`).

'Refresh Speedbar (r)'
> Regenerates the list of folders in the speedbar. Run this command if you've added or deleted a folder, or want to update the unseen message count before the next automatic update (`mh-speed-refresh`).

You can click on *Mouse-3* to bring up a context menu that contains these items. Dismiss the speedbar with *C-x 5 0* (`delete-frame`).

The MH-E speedbar uses the MH command `flists`[1] to generate the list of folders. The '`mh-speedbar`' customization group contains the following option which controls how often the speedbar calls `flists`.

`mh-speed-update-interval`
> Time between speedbar updates in seconds (default: 60). Set to 0 to disable automatic update.

You can modify the appearance of the folders in the speedbar by customizing the following faces.

`mh-speedbar-folder`
> Basic folder face.

`mh-speedbar-folder-with-unseen-messages`
> Folder face when folder contains unread messages.

`mh-speedbar-selected-folder`
> Selected folder face.

`mh-speedbar-selected-folder-with-unseen-messages`
> Selected folder face when folder contains unread messages.

---

[1] See the section Searching for Sequences with flist in the MH book.

13 The Menu Bar

For those of you who prefer to mouse and menu instead of using the meta-coke-bottle-bucky keys, MH-E provides menu items for most of its functions. The MH-Folder buffer adds the 'Folder', 'Message', and 'Sequence' menus. The MH-Letter buffer adds the 'Identity' and 'Letter' menus. The MH-Search buffer adds the 'Search' menu. There's no need to list the actual items here, as you can more easily see them for yourself, and the functions are already described elsewhere in this manual.

For a description of the menu bar, please See section "The Menu Bar" in *The GNU Emacs Manual*.

The Emacs manual describes how to get online help for a particular menu item. You can also look up a menu item in the index of this manual in two ways: all of the menu items are listed alphabetically, and you can also browse all of the items under the index entry 'menu item'.

14 The Tool Bar

Emacs also provides a graphical tool bar. For a description of the tool bar, please See section "Tool Bars" in *The GNU Emacs Manual*.

MH-E adds several icons to this tool bar; you can modify the MH-E aspects of the tool bar via the 'mh-tool-bar' customization group.

mh-tool-bar-folder-buttons
> List of buttons to include in MH-Folder tool bar (default: a checklist too long to list here).

mh-tool-bar-letter-buttons
> List of buttons to include in MH-Letter tool bar (default: a checklist too long to list here).

mh-tool-bar-search-function
> Function called by the tool bar search button (default: mh-search).

mh-xemacs-tool-bar-position
> Tool bar location (default: 'Same As Default Tool Bar').

mh-xemacs-use-tool-bar-flag
> If 'on', use tool bar (default: 'on', if supported).

In GNU Emacs, icons for some of MH-E's functions are added to the tool bar. In XEmacs, you have the opportunity to create a separate tool bar for the MH-E icons.

In either case, you can select which of these functions you'd like to see by customizing the options mh-tool-bar-folder-buttons and mh-tool-bar-letter-buttons. As you probably guessed, the former customizes the tool bar in MH-Folder mode and the latter in MH-Letter mode. Both of these options present you with a list of functions; check the functions whose icons you want to see and clear the check boxes for those you don't.

The function associated with the searching icon can be set via the option mh-tool-bar-search-function. By default, this is set to mh-search. See Chapter 15 [Searching], page 71. You can also choose 'Other Function' from the 'Value Menu' and enter a function of your own choosing.

XEmacs provides a couple of extra options. The first, mh-xemacs-use-tool-bar-flag, controls whether to show the MH-E icons at all. By default, this option is turned on if the window system supports tool bars. If your system doesn't support tool bars, then you won't be able to turn on this option.

The second extra option is mh-xemacs-tool-bar-position which controls the placement of the tool bar along the four edges of the frame. You can choose from one of 'Same As Default Tool Bar', 'Top', 'Bottom', 'Left', or 'Right'. If this variable is set to anything other than 'Same As Default Tool Bar' and the default tool bar is in a different location, then two tool bars will be displayed: the MH-E tool bar and the default tool bar.

15 Searching Through Messages

Earlier, the command *F s* (`mh-search`) was introduced which helps you find messages that lie buried in your folders (see Chapter 7 [Folders], page 33). This chapter covers this command in more detail. Several commands are used to compose the search criteria and to start searching. A couple of them can be found in the 'Search' menu.

C-c ?   Display cheat sheet for the MH-E commands (`mh-help`).

C-c C-c   Find messages using `mh-search-program` (`mh-index-do-search`).

C-c C-p   Find messages using `pick` (`mh-pick-do-search`).

C-c ?   Display cheat sheet for the MH-E commands (`mh-help`).

C-c C-f a
C-c C-f C-a
         Move to 'Mail-Reply-To:' header field (`mh-to-field`).

C-c C-f b
C-c C-f C-b
         Move to 'Bcc:' header field (`mh-to-field`).

C-c C-f c
C-c C-f C-c
         Move to 'Cc:' header field (`mh-to-field`).

C-c C-f d
C-c C-f C-d
         Move to 'Dcc:' header field (`mh-to-field`).

C-c C-f f
C-c C-f C-f
         Move to 'Fcc:' header field (`mh-to-field`).

C-c C-f l
C-c C-f C-l
         Move to 'Mail-Followup-To:' header field (`mh-to-field`).

C-c C-f m
C-c C-f C-m
         Move to 'From:' header field (`mh-to-field`).

C-c C-f r
C-c C-f C-r
         Move to 'Reply-To:' header field (`mh-to-field`).

C-c C-f s
C-c C-f C-s
         Move to 'Subject:' header field (`mh-to-field`).

C-c C-f t
C-c C-f C-t
         Move to 'To:' header field (`mh-to-field`).

Another few commands are available in the MH-Folder buffer resulting from a search.

⟨TAB⟩   Jump to the next folder marker (`mh-index-next-folder`).

S-⟨TAB⟩   Jump to the previous folder marker (`mh-index-previous-folder`).

v   Visit original folder from where the message at point was found (`mh-index-visit-folder`).

There is one option from the '`mh-search`' customization group used in searching.

`mh-search-program`
          Search program that MH-E shall use (default: '`Auto-detect`').

The following hook is available.

`mh-search-mode-hook`
          Hook run upon entry to `mh-search-mode` (default: `nil`).

The following face is available.

`mh-search-folder`
          Folder heading face in MH-Folder buffers created by searches.

The command *F s* (`mh-search-folder`) helps you find messages in your entire corpus of mail. You can search for messages to or from a particular person or about a particular subject. In fact, you can also search for messages containing selected strings in any arbitrary header field or any string found within the messages.

Out of the box, MH-E uses `pick` to find messages. With a little extra effort, you can set an indexing program which rewards you with extremely quick results. The drawback is that sometimes the index does not contain the words you're looking for. You can still use `pick` in these situations.

You are prompted for the folder to search. This can be '`all`' to search all folders. Note that the search works recursively on the listed folder.

Next, an MH-Search buffer appears where you can enter search criteria.

```
From:
To:
Cc:
Date:
Subject:
--------
#




--:**  search-pattern   All L7    (MH-Search)-------------------------
Type C-c C-c to search messages, C-c C-p to use pick, C-c ? for help
```

Search window

Edit this template by entering your search criteria in an appropriate header field that is already there, or create a new field yourself. If the string you're looking for could be anywhere in a message, then place the string underneath the row of dashes.

As an example, let's say that we want to find messages from Ginnean about horseback riding in the Kosciusko National Park (Australia) during January, 1994. Normally we would start with a broad search and narrow it down if necessary to produce a manageable amount of data, but we'll cut to the chase and create a fairly restrictive set of criteria as follows:

```
From: ginnean
To:
Cc:
Date: Jan 1994
Subject:
--------
horse
kosciusko
```

As with MH-Letter mode, MH-Search provides commands like `C-c C-f C-t` (`mh-to-field`) to help you fill in the blanks. See Section 9.1 [Editing Message], page 48.

If you find that you do the same thing over and over when editing the search template, you may wish to bind some shortcuts to keys. This can be done with the variable `mh-search-mode-hook`, which is called when `F s` is run on a new pattern.

To perform the search, type `C-c C-c` (`mh-index-do-search`). Sometimes you're searching for text that is either not indexed, or hasn't been indexed yet. In this case you can override the default method with the pick method by running the command `C-c C-p` (`mh-pick-do-search`).

The messages that are found are put in a temporary sub-folder of '`+mhe-index`' and are displayed in an MH-Folder buffer. This buffer is special because it displays messages from multiple folders; each set of messages from a given folder has a heading with the folder name. The appearance of the heading can be modified by customizing the face `mh-search-folder`. You can jump back and forth between the headings using the commands `TAB` (`mh-index-next-folder`) and `S-`TAB` (`mh-index-previous-folder`).

In addition, the command `v` (`mh-index-visit-folder`) can be used to visit the folder of the message at point. Initially, only the messages that matched the search criteria are displayed in the folder. While the temporary buffer has its own set of message numbers, the actual messages numbers are shown in the visited folder. Thus, the command `v` is useful to find the actual message number of an interesting message, or to view surrounding messages with the command `F r mh-rescan-folder`. See Chapter 7 [Folders], page 33.

Because this folder is temporary, you'll probably get in the habit of killing it when you're done with `F k` (`mh-kill-folder`). See Chapter 7 [Folders], page 33.

You can regenerate the results by running `F s` with a prefix argument.

Note: This command uses an '`X-MHE-Checksum:`' header field to cache the MD5 checksum of a message. This means that if an incoming message already contains an '`X-MHE-Checksum:`' field, that message might not be found by this command. The following `procmail` recipe avoids this problem by renaming the existing header field:

```
:0 wf
| formail -R "X-MHE-Checksum" "X-Old-MHE-Checksum"
```

See Chapter 17 [Limits], page 79, for an alternative interface to searching.

15.1 Configuring Indexed Searches

The command *F s* (`mh-search`) runs the command defined by the option `mh-search-program`. The default value is '`Auto-detect`' which means that MH-E will automatically choose one of `swish++`, `swish-e`, `mairix`, `namazu`, `pick` and `grep` in that order. If, for example, you have both `swish++` and `mairix` installed and you want to use `mairix`, then you can set this option to '`mairix`'.

The following sub-sections describe how to set up the various indexing programs to use with MH-E.

15.1.1 swish++

In the examples below, replace '`/home/user/Mail`' with the path to your MH directory.

First create the directory '`/home/user/Mail/.swish++`'. Then create the file '`/home/user/Mail/.swish++/swish++.conf`' with the following contents:

```
IncludeMeta        Bcc Cc Comments Content-Description From Keywords
IncludeMeta        Newsgroups Resent-To Subject To
IncludeMeta        Message-Id References In-Reply-To
IncludeFile        Mail    *
IndexFile          /home/user/Mail/.swish++/swish++.index
```

Use the following command line to generate the swish index. Run this daily from cron:

```
find /home/user/Mail -path /home/user/Mail/mhe-index -prune \
                     -o -path /home/user/Mail/.swish++ -prune \
                     -o -name "[0-9]*" -print \
      | index -c /home/user/Mail/.swish++/swish++.conf -
```

This command does not index the folders that hold the results of your searches in '`+mhe-index`' since they tend to be ephemeral and the original messages are indexed anyway.

On some systems (Debian GNU/Linux, for example), use `index++` instead of `index`.

15.1.2 swish

In the examples below, replace '`/home/user/Mail`' with the path to your MH directory.

First create the directory '`/home/user/Mail/.swish`'. Then create the file '`/home/user/Mail/.swish/config`' with the following contents:

```
DefaultContents TXT*
IndexDir /home/user/Mail
IndexFile /home/user/Mail/.swish/index
IndexName "Mail Index"
IndexDescription "Mail Index"
IndexPointer "http://nowhere"
IndexAdmin "nobody"
#MetaNames automatic
IndexReport 3
FollowSymLinks no
UseStemming no
IgnoreTotalWordCountWhenRanking yes
WordCharacters abcdefghijklmnopqrstuvwxyz0123456789-
BeginCharacters abcdefghijklmnopqrstuvwxyz
EndCharacters abcdefghijklmnopqrstuvwxyz0123456789
IgnoreLimit 50 1000
IndexComments 0
FileRules filename contains \D
FileRules pathname contains /home/user/Mail/.swish
FileRules pathname contains /home/user/Mail/mhe-index
FileRules filename is index
```

This configuration does not index the folders that hold the results of your searches in '+mhe-index' since they tend to be ephemeral and the original messages are indexed anyway.

If there are any directories you would like to ignore, append lines like the following to 'config':

```
FileRules pathname contains /home/user/Mail/scripts
```

Use the following command line to generate the swish index. Run this daily from cron:

```
swish-e -c /home/user/Mail/.swish/config
```

15.1.3 mairix

In the examples below, replace '/home/user/Mail' with the path to your MH directory.

First create the directory '/home/user/Mail/.mairix'. Then create the file '/home/user/Mail/.mairix/config' with the following contents:

```
base=/home/user/Mail

# List of folders that should be indexed. 3 dots at the end means there
# are subfolders within the folder
mh=archive...:inbox:drafts:news:sent:trash

vfolder_format=raw
database=/home/user/Mail/mairix/database
```

Use the following command line to generate the mairix index. Run this daily from cron:

```
mairix -f /home/user/Mail/.mairix/config
```

15.1.4 namazu

In the examples below, replace '/home/user/Mail' with the path to your MH directory.

First create the directory '/home/user/Mail/.namazu'. Then create the file '/home/user/Mail/.namazu/mknmzrc' with the following contents:

```
package conf;  # Don't remove this line!
$ADDRESS = 'user@localhost';
$ALLOW_FILE = "[0-9]*";
$EXCLUDE_PATH = "^/home/user/Mail/(mhe-index|spam)";
```

This configuration does not index the folders that hold the results of your searches in '`+mhe-index`' since they tend to be ephemeral and the original messages are indexed anyway.

Use the following command line to generate the namazu index. Run this daily from cron:

```
mknmz -f /home/user/Mail/.namazu/mknmzrc -O /home/user/Mail/.namazu \
        /home/user/Mail
```

15.1.5 pick

This search method does not require any setup.

Read `pick`(1) or the section Finding Messages with pick in the MH book to find out more about how to enter the criteria.

15.1.6 grep

This search method does not require any setup.

Unlike the other search methods, this method does not use the MH-Search buffer. Instead, you simply enter a regular expression in the minibuffer. For help in constructing regular expressions, see your man page for `grep`.

16 Viewing Message Threads

MH-E groups messages by *threads* which are messages that are part of the same discussion and usually all have the same '`Subject:`' header field. Other ways to organize messages in a folder include limiting (see Chapter 17 [Limits], page 79) or using full-text indexed searches (see Chapter 15 [Searching], page 71).

A thread begins with a single message called a *root*. All replies to the same message are *siblings* of each other. Any message that has replies to it is an *ancestor* of those replies.

There are several commands that you can use to navigate and operate on threads.

T ?   Display cheat sheet for the commands of the current prefix in minibuffer (`mh-prefix-help`).

T o   Refile (output) thread into folder (`mh-thread-refile`).

T d   Delete thread (`mh-thread-delete`).

T t   Toggle threaded view of folder (`mh-toggle-threads`).

T n   Display next sibling (`mh-thread-next-sibling`).

T p   Display previous sibling (`mh-thread-previous-sibling`).

T u   Display ancestor of current message (`mh-thread-ancestor`).

The '`mh-thread`' customization group contains one option.

`mh-show-threads-flag`
            On means new folders start in threaded mode (default: '`off`').

Threading large number of messages can be time consuming so the option `mh-show-threads-flag` is turned off by default. If you turn on this option, then threading will be done only if the number of messages being threaded is less than `mh-large-folder`. In any event, threading can be turned on (and off) with the command *T t* (`mh-toggle-threads`).

There are a few commands to help you navigate threads. If you do not care for the way a particular thread has turned, you can move up the chain of messages with the command *T u* (`mh-thread-ancestor`. At any point you can use *T n* (`mh-thread-next-sibling` or *T p* (`mh-thread-previous-sibling`) to jump to the next or previous sibling, skipping the sub-threads. The command *T u* can also take a prefix argument to jump to the message that started everything.

There are threaded equivalents for the commands that delete and refile messages. For example, *T o* (`mh-thread-refile`) refiles the current message and all its children. Similarly, the command *T d* (`mh-thread-delete`) deletes the current message and all its children. These commands do not refile or delete sibling messages. See Section 6.8 [Navigating], page 30, for a description of the similar command *k* (`mh-delete-subject-or-thread`).

If you find that threading is too slow, it may be that you have `mh-large-folder` set too high. Also, threading is one of the few features of MH-E that really benefits from compiling. If you haven't compiled MH-E, I encourage you to do so[1].

---

[1]   If you're not sure if MH-E has been byte-compiled, you could try running '`locate mh-thread.elc`' or otherwise find MH-E on your system and ensure that '`mh-thread.elc`' exists. If you have multiple

---

versions and you find that one is compiled but the other is not, then go into your '`*scratch*`' buffer in Emacs, enter `load-path C-j`, and ensure that the byte-compiled version appears first in the `load-path`. If you find that MH-E is not compiled and you installed MH-E yourself, please refer to the installation directions in the file '`README`' in the distribution.

17 Limiting Display

Another way to organize messages in a folder besides threading (see Chapter 16 [Threading], page 77) or using full-text indexed searches (see Chapter 15 [Searching], page 71) is by limiting the folder display to messages that are similar to the current message.

`/ ?`	Display cheat sheet for the commands of the current prefix in minibuffer (`mh-prefix-help`).
`/ '`	Limit to messages in the 'tick' sequence (`mh-narrow-to-tick`).
`/ c`	Limit to messages with the same 'Cc:' field (`mh-narrow-to-cc`).
`/ m`	Limit to messages with the same 'From:' field (`mh-narrow-to-from`).
`/ g`	Limit to range (`mh-narrow-to-range`).
`/ s`	Limit to messages with the same 'Subject:' field (`mh-narrow-to-subject`).
`/ t`	Limit to messages with the same 'To:' field (`mh-narrow-to-to`).
`/ w`	Remove last restriction (`mh-widen`).

All of the limiting commands above refine the display in some way.

The commands `/ c` (`mh-narrow-to-cc`), `/ m` (`mh-narrow-to-from`), `/ s` (`mh-narrow-to-subject`), and `/ t` (`mh-narrow-to-to`) restrict the display to messages matching the content of the respective field in the current message. However, you can give any of these a prefix argument to edit the `pick` expression used to narrow the view[1].

You can also limit the display to messages in the 'tick' sequence with the command `/ '` (`mh-narrow-to-tick`). See Chapter 18 [Sequences], page 81, for information on putting message into the 'tick' sequence. Use the `/ g` (`mh-narrow-to-range`) command to limit the display to messages in a range (see Section 4.2 [Ranges], page 12).

Each limit can be undone in turn with the `/ w` (`mh-widen`) command. Give this command a prefix argument to remove all limits.

---

[1] See `pick`(1) or the section Finding Messages with pick in the MH book.

18 Using Sequences

For the whole scoop on MH sequences, refer to 'mh-sequence'(5)[1]. As you've read, several of the MH-E commands can operate on a sequence, which is a shorthand for a range or group of messages. For example, you might want to forward several messages to a friend or colleague. Here's how to manipulate sequences. These commands are also available in the 'Sequence' menu.

'   Toggle tick mark of range (mh-toggle-tick).

S ?   Display cheat sheet for the commands of the current prefix in minibuffer (mh-prefix-help).

S '   Limit to ticked messages (mh-narrow-to-tick).

S d   Delete range from sequence (mh-delete-msg-from-seq).

S k   Delete sequence (mh-delete-seq).

S l   List all sequences in folder (mh-list-sequences).

S n   Restrict display to messages in sequence (mh-narrow-to-seq).

S p   Add range to sequence (mh-put-msg-in-seq).

S s   Display the sequences in which the current message appears (mh-msg-is-in-seq).

S w   Remove last restriction (mh-widen).

M-x mh-update-sequences
             Flush MH-E's state out to MH.

   The 'mh-sequences' customization group contains the options associated with sequences.

mh-refile-preserves-sequences-flag
             On means that sequences are preserved when messages are refiled (default: 'on').

mh-tick-seq
             The name of the MH sequence for ticked messages (default: ''tick').

mh-update-sequences-after-mh-show-flag
             On means flush MH sequences to disk after message is shown (default: 'on').

   The following hook is available.

mh-unseen-updated-hook
             Hook run after the unseen sequence has been updated (default: nil).

   To place a message in a sequence, use S p (mh-put-msg-in-seq). Give S p a range and you can add all the messages in a sequence to another sequence (for example, C-u S p SourceSequence ⟨RET⟩ DestSequence ⟨RET⟩, see Section 4.2 [Ranges], page 12).

   One specific use of the S p command is ' (mh-toggle-tick) which adds messages to the 'tick' sequence. This sequence can be viewed later with the F ' (mh-index-ticked-messages) command (see Chapter 7 [Folders], page 33).

---

[1] See the section More About Sequences in the MH book.

You can customize the option `mh-tick-seq` if you already use the 'tick' sequence for your own use. You can also disable all of the ticking functions by choosing the 'Disable Ticking' item but there isn't much advantage to that.

Once you've placed some messages in a sequence, you may wish to narrow the field of view to just those messages in the sequence you've created. To do this, use *S n* (`mh-narrow-to-seq`). You are prompted for the name of the sequence. What this does is show only those messages that are in the selected sequence in the MH-Folder buffer. In addition, it limits further MH-E searches to just those messages. To narrow the view to the messages in the 'tick' sequence, use *S '* (`mh-narrow-to-tick`). When you want to widen the view to all your messages again, use *S w* (`mh-widen`).

You can see which sequences in which a message appears with the command *S s* (`mh-msg-is-in-seq`). Use a prefix argument to display the sequences in which another message appears (as in *C-u 42 S s* RET). Or, you can list all sequences in a selected folder (default is current folder) with *S l* (`mh-list-sequences`). The list appears in a buffer named '*MH-E Sequences*' (see Chapter 20 [Miscellaneous], page 89).

If a message is in any sequence (except 'Previous-Sequence:'[2] and 'cur') when it is refiled, then it will still be in those sequences in the destination folder. If this behavior is not desired, then turn off the option `mh-refile-preserves-sequences-flag`.

If you want to remove a message (or range, see Section 4.2 [Ranges], page 12) from a sequence, use *S d* (`mh-delete-msg-from-seq`). If you want to delete an entire sequence, use *S k* (`mh-delete-seq`). In the latter case you are prompted for the sequence to delete. Note that this deletes only the sequence, not the messages in the sequence. If you want to delete the messages, use *C-u d* (see Chapter 6 [Reading Mail], page 17).

Three sequences are maintained internally by MH-E and pushed out to MH when a message is shown. They include the sequence specified by your 'Unseen-Sequence:' profile component, 'cur', and the sequence listed by the option `mh-tick-seq` which is 'tick' by default. If you do not like this behavior, turn off the option `mh-update-sequences-after-mh-show-flag`. You can then update the state manually with the *x*, *q*, or *M-x mh-update-sequences* commands.

The hook `mh-unseen-updated-hook` is run after the unseen sequence has been updated. The variable `mh-seen-list` can be used by this hook to obtain the list of messages which were removed from the unseen sequence.

With the exceptions of *S n* and *S w*, the underlying MH command dealing with sequences is `mark`[3].

---

[2] See 'mh-profile'(5)).

[3] See the section Make Message Bookmarks with mark in the MH book.

19 Dealing With Junk Mail

Marshall Rose once wrote a paper on MH entitled, *How to process 200 messages a day and still get some real work done.* This chapter could be entitled, *How to process 1000 spams a day and still get some real work done.*

We use the terms *junk mail* and *spam* interchangeably for any unwanted message which includes spam, *viruses*, and *worms*. The opposite of spam is *ham*. The act of classifying a sender as one who sends junk mail is called *blacklisting*; the opposite is called *whitelisting*.

J ?   Display cheat sheet for the commands of the current prefix in minibuffer (`mh-prefix-help`).

J b   Blacklist range as spam (`mh-junk-blacklist`).

J w   Whitelist range as ham (`mh-junk-whitelist`).

`mh-spamassassin-identify-spammers`
                 Identify spammers who are repeat offenders.

The following table lists the options from the '`mh-junk`' customization group.

`mh-junk-background`
                 If on, spam programs are run in background (default: '`off`').

`mh-junk-disposition`
                 Disposition of junk mail (default: '`Delete Spam`').

`mh-junk-program`
                 Spam program that MH-E should use (default: '`Auto-detect`').

MH-E depends on SpamAssassin, bogofilter, or SpamProbe to throw the dreck away. This chapter describes briefly how to configure these programs to work well with MH-E and how to use MH-E's interface that provides continuing education for these programs.

The default setting of the option `mh-junk-program` is '`Auto-detect`' which means that MH-E will automatically choose one of SpamAssassin, bogofilter, or SpamProbe in that order. If, for example, you have both SpamAssassin and bogofilter installed and you want to use bogofilter, then you can set this option to '`Bogofilter`'.

The command *J b* (`mh-junk-blacklist`) trains the spam program in use with the content of the range (see Section 4.2 [Ranges], page 12) and then handles the message(s) as specified by the option `mh-junk-disposition`. By default, this option is set to '`Delete Spam`' but you can also specify the name of the folder which is useful for building a corpus of spam for training purposes.

In contrast, the command *J w* (`mh-junk-whitelist`) reclassifies a range of messages (see Section 4.2 [Ranges], page 12) as ham if it were incorrectly classified as spam. It then refiles the message into the '`+inbox`' folder.

By default, the programs are run in the foreground, but this can be slow when junking large numbers of messages. If you have enough memory or don't junk that many messages at the same time, you might try turning on the option `mh-junk-background`.[1]

---

[1] Note that the option `mh-junk-background` is used as the `display` argument in the call to `call-process`. Therefore, turning on this option means setting its value to '`0`'. You can also set its value to '`t`' to direct the programs' output to the '`*MH-E Log*`' buffer; this may be useful for debugging.

The following sections discuss the various counter-spam measures that MH-E can work with.

SpamAssassin

SpamAssassin is one of the more popular spam filtering programs. Get it from your local distribution or from the SpamAssassin web site.

To use SpamAssassin, add the following recipes to '`~/.procmailrc`':

```
PATH=$PATH:/usr/bin/mh
MAILDIR=$HOME/`mhparam Path`

# Fight spam with SpamAssassin.
:0fw
| spamc

# Anything with a spam level of 10 or more is junked immediately.
:0:
* ^X-Spam-Level: ..........
/dev/null

:0:
* ^X-Spam-Status: Yes
spam/.
```

If you don't use `spamc`, use '`spamassassin -P -a`'.

Note that one of the recipes above throws away messages with a score greater than or equal to 10. Here's how you can determine a value that works best for you.

First, run '`spamassassin -t`' on every mail message in your archive and use `gnumeric` to verify that the average plus the standard deviation of good mail is under 5, the SpamAssassin default for "spam".

Using `gnumeric`, sort the messages by score and view the messages with the highest score. Determine the score which encompasses all of your interesting messages and add a couple of points to be conservative. Add that many dots to the '`X-Spam-Level:`' header field above to send messages with that score down the drain.

In the example above, messages with a score of 5-9 are set aside in the '`+spam`' folder for later review. The major weakness of rules-based filters is a plethora of false positives so it is worthwhile to check.

If SpamAssassin classifies a message incorrectly, or is unsure, you can use the MH-E commands _J b_ (`mh-junk-blacklist`) and _J w_ (`mh-junk-whitelist`).

The command _J b_ (`mh-junk-blacklist`) adds a '`blacklist_from`' entry to '`~/spamassassin/user_prefs`', deletes the message, and sends the message to the Razor, so that others might not see this spam. If the `sa-learn` command is available, the message is also recategorized as spam.

The command _J w_ (`mh-junk-whitelist`) adds a '`whitelist_from`' rule to '`~/.spamassassin/user_prefs`'. If the `sa-learn` command is available, the message is also recategorized as ham.

Over time, you'll observe that the same host or domain occurs repeatedly in the '`blacklist_from`' entries, so you might think that you could avoid future spam by blacklisting all mail from a particular domain. The utility function `mh-spamassassin-identify-spammers` helps you do precisely that. This function displays a frequency count

of the hosts and domains in the 'blacklist_from' entries from the last blank line in
'~/.spamassassin/user_prefs' to the end of the file. This information can be used so
that you can replace multiple 'blacklist_from' entries with a single wildcard entry such
as:

```
blacklist_from *@*amazingoffersdirect2u.com
```

In versions of SpamAssassin (2.50 and on) that support a Bayesian classifier, J b (mh-
junk-blacklist) uses the program sa-learn to recategorize the message as spam. Neither
MH-E, nor SpamAssassin, rebuilds the database after adding words, so you will need to
run 'sa-learn --rebuild' periodically. This can be done by adding the following to your
'crontab':

```
0 * * * * sa-learn --rebuild > /dev/null 2>&1
```

Bogofilter

Bogofilter is a Bayesian spam filtering program. Get it from your local distribution or from
the bogofilter web site.

Bogofilter is taught by running:

```
bogofilter -n < good-message
```

on every good message, and

```
bogofilter -s < spam-message
```

on every spam message. This is called a *full training*; three other training methods are
described in the FAQ that is distributed with bogofilter. Note that most Bayesian filters
need 1000 to 5000 of each type of message to start doing a good job.

To use bogofilter, add the following recipes to '~/.procmailrc':

```
PATH=$PATH:/usr/bin/mh
MAILDIR=$HOME/`mhparam Path`

# Fight spam with Bogofilter.
:0fw
| bogofilter -3 -e -p

:0:
* ^X-Bogosity: Yes, tests=bogofilter
spam/.

:0:
* ^X-Bogosity: Unsure, tests=bogofilter
spam/unsure/.
```

If bogofilter classifies a message incorrectly, or is unsure, you can use the MH-E com-
mands J b (mh-junk-blacklist) and J w (mh-junk-whitelist) to update bogofilter's
training.

The *Bogofilter FAQ* suggests that you run the following occasionally to shrink the data-
base:

```
bogoutil -d wordlist.db | bogoutil -l wordlist.db.new
mv wordlist.db wordlist.db.prv
mv wordlist.db.new wordlist.db
```

The *Bogofilter tuning HOWTO* describes how you can fine-tune bogofilter.

SpamProbe

SpamProbe is a Bayesian spam filtering program. Get it from your local distribution or from the SpamProbe web site.

To use SpamProbe, add the following recipes to '`~/.procmailrc`':

```
PATH=$PATH:/usr/bin/mh
MAILDIR=$HOME/`mhparam Path`

# Fight spam with SpamProbe.
:0
SCORE=| spamprobe receive

:0 wf
| formail -I "X-SpamProbe: $SCORE"

:0:
*^X-SpamProbe: SPAM
spam/.
```

If SpamProbe classifies a message incorrectly, you can use the MH-E commands *J b* (`mh-junk-blacklist`) and *J w* (`mh-junk-whitelist`) to update SpamProbe's training.

Other Things You Can Do

There are a couple of things that you can add to '`~/.procmailrc`' in order to filter out a lot of spam and viruses. The first is to eliminate any message with a Windows executable (which is most likely a virus). The second is to eliminate mail in character sets that you can't read.

```
PATH=$PATH:/usr/bin/mh
MAILDIR=$HOME/`mhparam Path`

#
# Filter messages with win32 executables/virii.
#
# These attachments are base64 and have a TVqQAAMAAAAEAAAA//8AALg
# pattern. The string "this program cannot be run in MS-DOS mode"
# encoded in base64 is 4fug4AtAnNIbg and helps to avoid false
# positives (Roland Smith via Pete from the bogofilter mailing list).
#
:0 B:
* ^Content-Transfer-Encoding:.*base64
* ^TVqQAAMAAAAEAAAA//8AALg
* 4fug4AtAnNIbg
spam/exe/.

#
# Filter mail in unreadable character sets (from the Bogofilter FAQ).
#
UNREADABLE='[^?"]*big5|iso-2022-jp|ISO-2022-KR|euc-kr|gb2312|ks_c_5601-1987'

:0:
* 1^0 $ ^Subject:.*=\?($UNREADABLE)
* 1^0 $ ^Content-Type:.*charset="?($UNREADABLE)
spam/unreadable/.

:0:
* ^Content-Type:.*multipart
```

```
* B ?? $ ^Content-Type:.*^?.*charset="?($UNREADABLE)
spam/unreadable/.
```

20 Miscellaneous Commands, Variables, and Buffers

This chapter covers the following command and the various MH-E buffers,

`mh-version`
>   Display version information about MH-E and the MH mail handling system.

One command worth noting is `M-x mh-version`. You can compare the version this command prints to the latest release (see Section C.4 [Getting MH-E], page 99). The output of `M-x mh-version`, found in a buffer named '`*MH-E Info*`', should usually be included with any bug report you submit (see Section C.1 [Bug Reports], page 99).

MH-E Buffers

Besides the MH-Folder, MH-Show, and MH-Letter buffers, MH-E creates several other buffers. They are:

'`*MH-E Folders*`'
>   This buffer contains the output of `F l` (`mh-list-folders`). See Chapter 7 [Folders], page 33.

'`*MH-E Help*`'
>   This buffer contains the output of `?` (`mh-help`) and `C-c ?` in MH-Letter mode. See Chapter 4 [Using This Manual], page 11.

'`*MH-E Info*`'
>   This buffer contains the output of `M-x mh-version` (RET).

'`*MH-E Log*`'
>   This buffer contains the last 100 lines of the output of the various MH commands.

'`*MH-E Mail Delivery*`'
>   This buffer contains the transcript of a mail delivery. See Section 9.9 [Sending Message], page 56.

'`*MH-E Recipients*`'
>   This buffer contains the output of `C-c C-w` (`mh-check-whom`) and is killed when draft is sent. See Section 9.8 [Checking Recipients], page 56.

'`*MH-E Sequences*`'
>   This buffer contains the output of `S l` (`mh-list-sequences`). See Chapter 18 [Sequences], page 81.

'`*mh-temp*`'
>   This is a scratch, ephemeral, buffer used by MH-E functions. Note that it is hidden because the first character in the name is a space. You'll generally not have any need for this buffer.

Appendix A Scan Line Formats

This appendix discusses how MH-E creates, parses, and manipulates scan lines. If you have your own MH scan or inc format files, you **can** teach MH-E how to handle them, but it isn't easy as you'll see.

This table lists the options in the 'mh-scan-line-formats' customization group.

mh-adaptive-cmd-note-flag

On means that the message number width is determined dynamically (default: 'on').

mh-scan-format-file

Specifies the format file to pass to the scan program (default: 'Use MH-E scan Format').

mh-scan-prog

Program used to scan messages (default: "scan").

There are a couple of caveats when creating your own scan format file. First, MH-E will not work if your scan lines do not include message numbers. It will work poorly if you don't dedicate a column for showing the current message and notations. You won't be able to use the option mh-adaptive-cmd-note-flag or the threading features (see Chapter 16 [Threading], page 77).

If you've created your own format to handle long message numbers, you'll be pleased to know you no longer need it since MH-E adapts its internal format based upon the largest message number if mh-adaptive-cmd-note-flag is on (the default). If you prefer fixed-width message numbers, turn off mh-adaptive-cmd-note-flag and call mh-set-cmd-note with the width specified by your format file (see mh-scan-format-file). For example, the default width is 4, so you would use '(mh-set-cmd-note 4)'.

The default setting for mh-scan-format-file is 'Use MH-E scan Format'. This means that the format string will be taken from the either mh-scan-format-mh or mh-scan-format-nmh depending on whether MH or nmh (or GNU mailutils) is in use. This setting also enables you to turn on the option mh-adaptive-cmd-note-flag. You can also set this option to 'Use Default scan Format' to get the same output as you would get if you ran scan from the shell. If you have a format file that you want MH-E to use but not MH, you can set this option to 'Specify a scan Format File' and enter the name of your format file.

The scan format that MH-E uses when mh-scan-format-file is set to its default of 'Use MH-E scan Format' is held in the variables mh-scan-format-nmh and mh-scan-format-mh depending on whether you are using nmh (or GNU mailutils) or not. Typically, you create your own format files rather than modifying these variables. The value of mh-scan-format-nmh is:

```
(concat
 "%4(msg)"
 "%<(cur)+%| %>"
 "%<{replied}-"
 "%?(nonnull(comp{to}))%<(mymbox{to})t%>"
 "%?(nonnull(comp{cc}))%<(mymbox{cc})c%>"
 "%?(nonnull(comp{bcc}))%<(mymbox{bcc})b%>"
```

```
"%?(nonnull(comp{newsgroups}))n%>"
"%<(zero) %>"
"%02(mon{date})/%02(mday{date})%<{date} %|*%>"
"%<(mymbox{from})%<{to}To:%14(decode(friendly{to}))%>%>"
"%<(zero)%17(decode(friendly{from}))%>  "
"%(decode{subject})%<{body}<<%{body}%>")
```

The setting for `mh-scan-format-mh` is similar, except that MH doesn't have the function `decode` (which is used to decode RFC 2047 encodings).

These strings are passed to the `scan` program via the '`-format`' argument. The formats are identical to the defaults except that additional hints for fontification have been added to the existing notations in the fifth column (remember that in Emacs, the columns start at 0). The values of the fifth column, in priority order, are: '`-`' if the message has been replied to, '`t`' if an address in the '`To:`' field matches one of the mailboxes of the current user, '`c`' if the '`Cc:`' field matches, '`b`' if the '`Bcc:`' field matches, and '`n`' if a non-empty '`Newsgroups:`' field is present.

The name of the program that generates a listing of one line per message is held in `mh-scan-prog` (default: `"scan"`). Unless this variable contains an absolute pathname, it is assumed to be in the `mh-progs` directory (see Chapter 2 [Getting Started], page 5). You may link another program to `scan` (see '`mh-profile`'(5)) to produce a different type of listing[1].

If you change the format of the scan lines you'll need to tell MH-E how to parse the new format. As you will see, quite a lot of variables are involved to do that. Use `M-x apropos` ⟨RET⟩ `mh-scan.*regexp` ⟨RET⟩ to obtain a list of these variables. You will also have to call `mh-set-cmd-note` if your notations are not in column 4 (columns in Emacs start with 0). Note that unlike most of the user options described in this manual, these are variables and must be set with `setq` instead of in a customization buffer. For help with regular expressions, see section "Syntax of Regular Expressions" in The GNU Emacs Manual.

The first variable has to do with pruning out garbage.

`mh-scan-valid-regexp`

> This regular expression describes a valid scan line. This is used to eliminate error messages that are occasionally produced by `inc`[2] or `scan` (default: `"^ *[0-9]"`).

Next, many variables control how the scan lines are parsed.

`mh-scan-body-regexp`

> This regular expression matches the message body fragment. Note that the default setting of `mh-folder-font-lock-keywords` expects this expression to contain at least one parenthesized expression which matches the body text as in the default of `"\\(<<\\([^\n]+\\)?\\)"`. If this regular expression is not correct, the body fragment will not be highlighted with the face `mh-folder-body`.

`mh-scan-cur-msg-number-regexp`

> This regular expression matches the current message. It must match from the beginning of the line. Note that the default setting of `mh-folder-`

---

[1] See the section Find and Specify with scan pick Ranges Sequences in the MH book.

[2] See the section Reading Mail: inc show next prev in the MH book.

`font-lock-keywords` expects this expression to contain at least one parenthesized expression which matches the message number as in the default of `"^\\( *[0-9]+\\+\\).*"`. This expression includes the leading space and current message marker '+' within the parenthesis since it looks better to highlight these items as well. The highlighting is done with the face `mh-folder-cur-msg-number`. This regular expression should be correct as it is needed by non-fontification functions. See also `mh-note-cur`.

`mh-scan-date-regexp`

This regular expression matches a valid date. It must **not** be anchored to the beginning or the end of the line. Note that the default setting of `mh-folder-font-lock-keywords` expects this expression to contain only one parenthesized expression which matches the date field as in the default of `"\\([0-9][0-9]/[0-9][0-9]\\)"`. If this regular expression is not correct, the date will not be highlighted with the face `mh-folder-date`.

`mh-scan-deleted-msg-regexp`

This regular expression matches deleted messages. It must match from the beginning of the line. Note that the default setting of `mh-folder-font-lock-keywords` expects this expression to contain at least one parenthesized expression which matches the message number as in the default of `"^\\( *[0-9]+\\)D"`. This expression includes the leading space within the parenthesis since it looks better to highlight it as well. The highlighting is done with the face `mh-folder-deleted`. This regular expression should be correct as it is needed by non-fontification functions. See also `mh-note-deleted`.

`mh-scan-good-msg-regexp`

This regular expression matches "good" messages. It must match from the beginning of the line. Note that the default setting of `mh-folder-font-lock-keywords` expects this expression to contain at least one parenthesized expression which matches the message number as in the default of `"^\\( *[0-9]+\\)[^D^0-9]"`. This expression includes the leading space within the parenthesis since it looks better to highlight it as well. The highlighting is done with the face `mh-folder-msg-number`. This regular expression should be correct as it is needed by non-fontification functions.

`mh-scan-msg-format-regexp`

This regular expression finds the message number width in a scan format. Note that the message number must be placed in a parenthesized expression as in the default of `"%\\([0-9]*\\)(msg)"`. This variable is only consulted if `mh-scan-format-file` is set to 'Use MH-E scan Format'.

`mh-scan-msg-format-string`

This is a format string for the width of the message number in a scan format. Use '0%d' for zero-filled message numbers. This variable is only consulted if `mh-scan-format-file` is set to 'Use MH-E scan Format' (default: `"%d"`).

`mh-scan-msg-number-regexp`

This regular expression extracts the message number. It must match from the beginning of the line. Note that the message number must be placed in a parenthesized expression as in the default of `"^ *\\([0-9]+\\)"`.

`mh-scan-msg-overflow-regexp`

> This regular expression matches overflowed message numbers (default: `"^[?0-9][0-9]"`).

`mh-scan-msg-search-regexp`

> This regular expression matches a particular message. It is a format string; use '`%d`' to represent the location of the message number within the expression as in the default of `"^[^0-9]*%d[^0-9]"`.

`mh-scan-rcpt-regexp`

> This regular expression specifies the recipient in messages you sent. Note that the default setting of `mh-folder-font-lock-keywords` expects this expression to contain two parenthesized expressions. The first is expected to match the 'To:' that the default scan format file generates. The second is expected to match the recipient's name as in the default of `"\\(To:\\)\\(.............\\)"`. If this regular expression is not correct, the 'To:' string will not be highlighted with the face `mh-folder-to` and the recipient will not be highlighted with the face `mh-folder-address`.

`mh-scan-refiled-msg-regexp`

> This regular expression matches refiled messages. It must match from the beginning of the line. Note that the default setting of `mh-folder-font-lock-keywords` expects this expression to contain at least one parenthesized expression which matches the message number as in the default of `"^\\( *[0-9]+\\)\\^"`. This expression includes the leading space within the parenthesis since it looks better to highlight it as well. The highlighting is done with the face `mh-folder-refiled`. This regular expression should be correct as it is needed by non-fontification functions. See also `mh-note-refiled`.

`mh-scan-sent-to-me-sender-regexp`

> This regular expression matches messages sent to us. Note that the default setting of `mh-folder-font-lock-keywords` expects this expression to contain at least two parenthesized expressions. The first should match the fontification hint (see `mh-scan-format-nmh`) and the second should match the user name as in the default of `"^ *[0-9]+.\\([bct]\\).....[ ]*\\(..................\\)"`. If this regular expression is not correct, the notation hints will not be highlighted with the face `mh-mh-folder-sent-to-me-hint` and the sender will not be highlighted with the face `mh-folder-sent-to-me-sender`.

`mh-scan-subject-regexp`

> This regular expression matches the subject. It must match from the beginning of the line. Note that the default setting of '`mh-folder-font-lock-keywords`' expects this expression to contain at least three parenthesized expressions. The first is expected to match the 'Re:' string, if any, and is highlighted with the face `mh-folder-followup`. The second matches an optional bracketed number after 'Re:', such as in 'Re[2]:' (and is thus a sub-expression of the first expression). The third is expected to match the subject line itself which is highlighted with the face `mh-folder-subject`. For example, the default is `"^ *[0-9]+........[ ]*...................`

`\\([Rr][Ee]\\(\\[[0-9]+\\]\\)?:\\s-*\\)*\\([^<\n]*\\)"`. This regular
expression should be correct as it is needed by non-fontification functions. Note
that this example is broken up on two lines for readability, but is actually a
single string.

Finally, there are a slew of variables that control how MH-E annotates the scan lines.

`mh-cmd-note`

Column for notations (default: 4). This variable should be set with the function
`mh-set-cmd-note`. This variable may be updated dynamically if `mh-adaptive-cmd-note-flag` is on. The following variables contain the notational characters.
Note that columns in Emacs start with 0.

`mh-note-copied`

Messages that have been copied are marked by this character (default: `?C`).

`mh-note-cur`

The current message (in MH, not in MH-E) is marked by this character (default:
`?+`). See also `mh-scan-cur-msg-number-regexp`.

`mh-note-deleted`

Messages that have been deleted are marked by this character (default: `?D`).
See also `mh-scan-deleted-msg-regexp`.

`mh-note-dist`

Messages that have been redistributed are marked by this character (default:
`?R`).

`mh-note-forw`

Messages that have been forwarded are marked by this character (default: `?F`).

`mh-note-printed`

Messages that have been printed are marked by this character (default: `?P`).

`mh-note-refiled`

Messages that have been refiled are marked by this character (default: `?^`). See
also `mh-scan-refiled-msg-regexp`.

`mh-note-repl`

Messages that have been replied to are marked by this character (default: `?-`).

`mh-note-seq`

Messages in a user-defined sequence are marked by this character (default: `?%`).
Messages in the 'search' sequence are marked by this character as well.

For example, let's say I have the following in 'scan.format' which displays the sender,
the subject, and the message number. This format places a '+' after the message number
for the current message according to MH; it also uses that column for notations.

```
%20(decode(friendly{from})) %50(decode{subject})  %4(msg)%<(cur)+%| %>
```

The first thing you have to do is tell MH-E to use this file. Customize `mh-scan-format-file` and set its value to 'Use Default scan Format'. If you didn't get already turn off
`mh-adaptive-cmd-note-flag`, you'll need to do that first.

Next, tell MH-E what a valid scan line looks like so that you can at least display the
output of scan in your MH-Folder buffer.

```
(setq mh-scan-valid-regexp "[0-9]+[+D^ ]$")
```

Now, in order to get rid of the 'Cursor not pointing to message' message, you need to tell MH-E how to access the message number. You should also see why MH-E requires that you include a message number in the first place.

```
(setq mh-scan-msg-number-regexp "^.* \\([0-9]+\\)[+D^ ]$")
(setq mh-scan-msg-search-regexp " %d[+D^ ]$")
```

In order to get the next and previous commands working, add this.

```
(setq mh-scan-good-msg-regexp "^.* \\([0-9]+\\)[+D^ ]$")
```

Note that the current message isn't marked with a '+' when moving between the next and previous messages. Here is the code required to get this working.

```
(set-mh-cmd-note 76)
(setq mh-scan-cur-msg-number-regexp "^.* \\([0-9]+\\)\\+$")
```

Finally, add the following to delete and refile messages.

```
(setq mh-scan-deleted-msg-regexp "^.* \\([0-9]+\\)D$")
(setq mh-scan-refiled-msg-regexp "^.* \\([0-9]+\\)\\^$")
```

This is just a bare minimum; it's best to adjust all of the regular expressions to ensure that MH-E and highlighting perform well.

Appendix B Reading Mailing Lists Effectively

This appendix explains how to use procmail to file mail from mailing lists into folders which can then be read easily with MH-E[1]. Some mailing lists have such high traffic that Gnus must be used and I discuss how to use Gnus side-by-side with MH-E.

First, I'll describe how to put mail from your mailing lists directly into an MH folder using procmail. First, add the following to '~/.procmailrc'. While the logging variables aren't strictly necessary, they are extremely useful.

```
[1]  # Update PATH so procmail can find myrcvstore, rcvstore and mhparam.
[2]  PATH=$PATH:/usr/lib/mh:/usr/bin/mh:$HOME/bin
[3]
[4]  # Point LOGFILE at the actual log file.
[5]  LOGFILE=$HOME/.procmail.log
[6]
[7]  # This setting provides just the right amount of information.
[8]  LOGABSTRACT=all
[9]
[10] # Uncomment the following line to see how your patterns match.
[11] #VERBOSE=yes
[12]
[13] # Place mail sent to any MH-E mailing list in +mh-e.
[14] :0 w: mh-e$LOCKEXT
[15] * ^TO.*mh-e-.*.*sourceforge.net
[16] | myrcvstore -create +mh-e
```

Line 14 creates a lock file in your mail directory based upon the name of the folder. This is done because rcvstore does not perform locking. While this lock file will prevent procmail from writing to a folder concurrently, there is a slight chance that you might lose a message if you're performing operations on a folder at the same time rcvstore is placing a message there. You have been warned. Now that that disclaimer is out of the way, note that I've been using this set-up for over a decade and haven't lost anything to my knowledge[2].

Line 16 uses the following script, myrcvstore, to massage the message as described in the comment and file the message in the given folder[3].

```
#! /bin/sh

# Accepts a message on standard input and passes it through rcvstore
# after first passing it through any filters. All arguments are passed
# on to rcvstore.

# Force the "From user date" to become part of header. One reason this
# is done is because the presence of the From field confuses dist so
# that dist adds a new header, rather than using the existing header.
# Note that this should not be done for any message that goes into a
# Gnus incoming file (Gnus will thrown an error) nor should it be
# applied to any message that goes to the system mailbox because the
# entire mailbox will be incorporated as a single message.
formail -c -z -R 'From ' X-Envelope-From: |
```

---

[1] The MH equivalent, slocal, can be used as well, but procmail is more flexible and more packages exist for procmail than for slocal.

[2] See Savannah issue #4361 to see if rcvstore locking is still an issue.

[3] The '-create' argument wasn't always the default to rcvstore.

```
rcvstore $@
```

If your version of `rcvstore` doesn't add messages to the 'unseen' sequence by default, add the following line to your MH profile:

```
Unseen-Sequence: unseen
```

Now view your new messages with the speedbar (see Chapter 12 [Speedbar], page 65) or with *F n* (`mh-index-new-messages`). See Chapter 7 [Folders], page 33.

If you're on a mailing list that is so voluminous that it is impossible to read every message, it usually better to read the mailing list like a newsgroup in a news reader. Emacs has a built-in newsreader called Gnus. The remainder of this appendix talks about how to use Gnus with an MH message store. The version of Gnus that was used to prepare this manual was 5.10. Versions 5.8 through 5.10 should work but versions prior to 5.8 use different options.

This table contains a list of Gnus options that you will have to modify. Note that for them to become accessible, you'll have to load '`nnml.el`' first. This can be done with *M-x load-library* ⟨RET⟩ *nnml* ⟨RET⟩.

`gnus-secondary-select-methods`

> Select the '`nnml`' value. This select method uses directories for folders and individual files for messages, just like MH. You do not have to set an address.

`mail-sources`

> Select the '`Several files in a directory`' value, check the '`Path`' box and enter '`~/Mail`' to tell Gnus where to find your mail.

`message-mail-user-agent`

> In order to send mail within Gnus using MH-E, set this option to '`mail-user-agent`' and set the `mail-user-agent` option to '`Emacs interface to MH`'.

`nnmail-keep-last-article`

> Since Gnus keeps track of which messages you have read, it would be bad if Gnus expired the last message, for example, message 100, and `rcvstore` gave the next new message number 1. Gnus would then ignore it since it thinks that you've read messages 1-100. Turning on this option ensures that the last message is never removed thereby eliminating this problem.

Next add the following to '`~/.procmailrc`'. If you don't subscribe to the GnuCash mailing list, substitute one to which you are subscribed.

```
PATH=$PATH:/usr/bin/mh
MAILDIR=$HOME/`mhparam Path`
# Place mail sent to the GnuCash mailing list in gnucash.spool, where
# Gnus will pick it up.
:0:
* ^TO.*gnucash.*.*gnucash.org
gnucash.spool
```

Wait for some messages to appear in '`gnucash.spool`' and run Gnus with *M-x gnus* ⟨RET⟩. To view the folder created in the example above, you would tell Gnus about it the first time only with *G m gnucash* ⟨RET⟩ *nnml* ⟨RET⟩. In MH-E, this folder is known as '`+gnucash`'.

Appendix C Odds and Ends

This appendix covers a few topics that don't fit elsewhere. Here I tell you how to report bugs and how to get on the MH-E mailing lists. I also point out some additional sources of information.

C.1 Bug Reports

Bug reports should be filed at SourceForge. You need to be a SourceForge user to submit bug reports, but this is easy enough to do that it shouldn't be a restriction for you. Please include the output of `M-x mh-version` (see Chapter 20 [Miscellaneous], page 89) in any bug report you send unless you're 110% positive we won't ask for it.

C.2 MH-E Mailing Lists

There are several mailing lists for MH-E. They are *mh-e-users at lists.sourceforge.net*, *mh-e-announce at lists.sourceforge.net*, and *mh-e-devel at lists.sourceforge.net*. You can subscribe or view the archives at SourceForge. Do not report bugs on these lists; please submit them via SourceForge (see Section C.1 [Bug Reports], page 99).

C.3 MH FAQ and Support

The article *MH Frequently Asked Questions (FAQ) with Answers* appears monthly in the newsgroup '`comp.mail.mh`'. While very little is there that deals with MH-E specifically, there is an incredible wealth of material about MH itself which you will find useful.

You can find FAQs on MH-E at the Support Requests page on SourceForge. If you don't find the answer to your question, file a support request and your question will become a new FAQ!

C.4 Getting MH-E

Because MH-E is undergoing a phase of sustained growth, the version of MH-E in your Emacs is likely to be out of date although it is most likely to be more up to date than the copy that comes with the MH distribution in '`miscellany/mh-e`'.

New MH-E releases are always available for downloading at SourceForge before they appear in an Emacs release. You can read the release notes on that page to determine if the given release of MH-E is already installed in your version of Emacs. You can also read the change log to see if you are interested in what the given release of MH-E has to offer (although we have no doubt that you will be extremely interested in all new releases).

If you use Debian, you can install the Debian mh-e package instead.

After you download and extract the MH-E tarball, read the '`README`' file and '`MH-E-NEWS`'. These correspond to the release notes and change log mentioned above. The file '`README`' contains instructions on installing MH-E. If you're already running Emacs, please quit that session and start again to load in the new MH-E. Check that you're running the new version with the command `M-x mh-version`.

In addition to the mh-e package, the SourceForge site also contains doc and contrib packages. The former is the latest release of this manual, and the latter contains a few contributed packages you might find useful.

Appendix D History of MH-E

MH-E was originally written by Brian Reid in 1983 and has changed hands several times since then. Jim Larus wanted to do something similar for GNU Emacs, and ended up completely rewriting it that same year. In 1989, Stephen Gildea picked it up and added many improvements. Bill Wohler then took over in 2000 and moved its development to SourceForge where it lives today.

D.1 From Brian Reid

One day in 1983 I got the flu and had to stay home from work for three days with nothing to do. I used that time to write MHE. The fundamental idea behind MHE was that it was a "puppeteer" driving the MH programs underneath it. MH had a model that the editor was supposed to run as a sub-process of the mailer, which seemed to me at the time to be the tail wagging the dog. So I turned it around and made the editor drive the MH programs. I made sure that the UCI people (who were maintaining MH at the time) took in my changes and made them stick.

Today, I still use my own version of MHE because I don't at all like the way that GNU MH-E works and I've never gotten to be good enough at hacking Emacs Lisp to make GNU MH-E do what I want. The Gosling-emacs version of MHE and the GNU Emacs version of MH-E have almost nothing in common except similar names. They work differently, have different conceptual models, and have different key bindings[1].

Brian Reid, June 1994

D.2 From Jim Larus

Brian Reid, while at CMU or shortly after going to Stanford wrote a mail reading program called MHE for Gosling Emacs. It had much the same structure as MH-E (i.e., invoked MH programs), though it was simpler and the commands were slightly different. Unfortunately, I no longer have a copy so the differences are lost in the mists of time.

In '82-83, I was working at BBN and wrote a lot of mlisp code in Gosling Emacs to make it look more like Tennex Emacs. One of the packages that I picked up and improved was Reid's mail system. In '83, I went back to Berkeley. About that time, Stallman's first version of GNU Emacs came out and people started to move to it from Gosling Emacs (as I recall, the transition took a year or two). I decided to port Reid's MHE and used the mlisp to Emacs Lisp translator that came with GNU Emacs. It did a lousy job and the resulting code didn't work, so I bit the bullet and rewrote the code by hand (it was a lot smaller and simpler then, so it took only a day or two).

Soon after that, MH-E became part of the standard Emacs distribution and suggestions kept dribbling in for improvements. MH-E soon reached sufficient functionality to keep me happy, but I kept on improving it because I was a graduate student with plenty of time on my hands and it was more fun than my dissertation. In retrospect, the one thing that I

---

[1] After reading this article, I questioned Brian about his version of MHE, and received some great ideas for improving MH-E such as a dired-like method of selecting folders; and removing the prompting when sending mail, filling in the blanks in the draft buffer instead. I passed them on to Stephen Gildea, the current maintainer, and he was excited about the ideas as well. Perhaps one day, MH-E will again resemble MHE (draft form editing was introduced in version 7.4).

regret is not writing any documentation, which seriously limited the use and appeal of the package.

In '89, I came to Wisconsin as a professor and decided not to work on MH-E. It was stable, except for minor bugs, and had enough functionality, so I let it be for a few years. Stephen Gildea of BBN began to pester me about the bugs, but I ignored them. In 1990, he went off to the X Consortium, said good bye, and said that he would now be using `xmh`. A few months later, he came back and said that he couldn't stand `xmh` and could I put a few more bug fixes into MH-E. At that point, I had no interest in fixing MH-E, so I gave the responsibility of maintenance to him and he has done a fine job since then.

Jim Larus, June 1994

D.3 From Stephen Gildea

In 1987 I went to work for Bolt Beranek and Newman, as Jim had before me. In my previous job, I had been using RMAIL, but as my folders tend to run large, I was frustrated with the speed of RMAIL. However, I stuck with it because I wanted the GNU Emacs interface. I am very familiar and comfortable with the Emacs interface (with just a few modifications of my own) and dislike having to use applications with embedded editors; they never live up to Emacs.

MH is the mail reader of choice at BBN, so I converted to it. Since I didn't want to give up using an Emacs interface, I started using MH-E. As is my wont, I started hacking on it almost immediately. I first used version 3.4m. One of the first features I added was to treat the folder buffer as a file-visiting buffer: you could lock it, save it, and be warned of unsaved changes when killing it. I also worked to bring its functionality a little closer to RMAIL. Jim Larus was very cooperative about merging in my changes, and my efforts first appeared in version 3.6, distributed with Emacs 18.52 in 1988. Next I decided MH-E was too slow and optimized it a lot. Version, 3.7, distributed with Emacs 18.56 in 1990, was noticeably faster.

When I moved to the X Consortium I became the first person there to not use xmh. (There is now one other engineer there using MH-E.) About this point I took over maintenance of MH-E from Jim and was finally able to add some features Jim hadn't accepted, such as the backward searching undo. My first release was 3.8 (Emacs 18.58) in 1992.

Now, in 1994, we see a flurry of releases, with both 4.0 and 5.0. Version 4.0 added many new features, including background folder collection and support for composing MIME messages. (Reading MIME messages remains to be done, alas.) While writing this book, Bill Wohler gave MH-E its closest examination ever, uncovering bugs and inconsistencies that required a new major version to fix, and so version 5 was released.

Stephen Gildea, June 1994

D.4 From Bill Wohler

The preface originally included the following text which I use to begin my story:

> But it's important to note a brief history of MH-E.

> Version 3 was prevalent through the Emacs 18 and early Emacs 19 years. Then Version 4 came out (Emacs 19.23), which introduced several new and changed

commands. Next, Version 5.0 was released, which fixed some bugs and incompatibilities, and was incorporated into Emacs 19.29.

After a long break, Stephen handed the reins over to me in 2000. I moved the project to a new site called SourceForge and organized a great team of developers. Our first release in late 2001 was version 6. It appeared around the time of Emacs 21.2 and had menus and tool bar buttons.

Then, indexed searches, improved MIME handling, a speedbar, multiple identities, alias completion, an index view of unseen messages, spam software support, Face and X-Image-URL header field support, Fcc completion, arbitrary range handling, and draft form editing were introduced in the version 7 series around the time of Emacs 21.4 (2004). Still, Emacs itself contained version 5 of MH-E released back in 1994.

Version 8 development was mostly driven by the rewrite of the manual. It also brought mailutils support, S/MIME support, picon support, and an improved interface for hiding header fields. The CVS repository was migrated from SourceForge to Savannah (only for those files that were already part of Emacs) and the software was completely reorganized to push back two decades of entropy. Version 8 will appear in Emacs 22.1, expected to be released in 2006.

Bill Wohler, February 2006

Appendix E GNU FREE DOCUMENTATION LICENSE

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque."

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements." Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at

your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties–for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications." You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted

document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled ``GNU
Free Documentation License''.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

```
with the Invariant Sections being list their titles, with the
Front-Cover Texts being list, and with the Back-Cover Texts being
list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Appendix F GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright © 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change
it. By contrast, the GNU General Public License is intended to guarantee your freedom
to share and change free software—to make sure the software is free for all its users. This
General Public License applies to most of the Free Software Foundation's software and to
any other program whose authors commit to using it. (Some other Free Software Foundation
software is covered by the GNU Lesser General Public License instead.) You can apply it
to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General
Public Licenses are designed to make sure that you have the freedom to distribute copies
of free software (and charge for this service if you wish), that you receive source code or
can get it if you want it, that you can change the software or use pieces of it in new free
programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you
these rights or to ask you to surrender the rights. These restrictions translate to certain
responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you
must give the recipients all the rights that you have. You must make sure that they, too,
receive or can get the source code. And you must show them these terms so they know
their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this
license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone
understands that there is no warranty for this free software. If the software is modified by
someone else and passed on, we want its recipients to know that what they have is not the
original, so that any problems introduced by others will not reflect on the original authors'
reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid
the danger that redistributors of a free program will individually obtain patent licenses, in
effect making the program proprietary. To prevent this, we have made it clear that any
patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program," below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification.") Each licensee is addressed as "you."

   Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

   These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a. Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

   If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

   It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

   This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

   Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version," you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software

which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
one line to give the program's name and an idea of what it does.
Copyright (C) yyyy  name of author

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) yyyy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details
type 'show w'.  This is free software, and you are welcome
to redistribute it under certain conditions; type 'show c'
for details.
```

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright
interest in the program 'Gnomovision'
(which makes passes at compilers) written
by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

Key (Character) Index

Command Index

N

O

P

S

T

Option (Variable) Index

Concept Index