IMAIL User's Manual

by Chris Hanson

This manual documents IMAIL 1.19.

Table of Contents

1 Introduction

IMAIL is a program for reading electronic mail. It uses the *Internet Message Access Protocol* (IMAP, RFC 2060) to access mail that is stored on a server, from which IMAIL fetches individual messages on demand. The server may have many different *folders* in which messages are stored, arranged in a hierarchical structure like that of a file system. Messages are easily moved or copied from one folder to another.

IMAP also supports the *Multipurpose Internet Mail Extensions* (MIME, RFC 2045), which facilitate the sending and receiving of *attachments*. The IMAP protocol supports this by allowing you to fetch some parts of a mail message while leaving others on the server. So, for example, if you receive a message containing a large attachment, it is possible to view the text of the message without waiting for the attachment to be fetched from the server; the attachment is fetched only if you want to view or save it. If you aren't interested in the attachment, you can delete the message without ever fetching it from the server.

In addition to these features, IMAIL provides a user interface very similar to that of the Emacs Rmail mail reader (see Section "Rmail" in *the GNU Emacs Manual*). IMAIL supports most of the same commands and has most of the same key bindings as Rmail. IMAIL is primarily intended to be an Rmail replacement for people who wish to read their mail using an IMAP server. IMAIL can also read and write Rmail files and unix mail (mbox) files, and provides the ability to copy messages from such a file to an IMAP folder, or vice versa; this greatly simplifies the transition from Rmail to IMAIL for those of us who have large amounts of mail stored in files.

2 Getting Started

At present, IMAIL has only a very simple mechanism for connecting to an IMAP server: it makes an unencrypted connection to the server, and logs in with a user name and a password. In the near future, we will implement CRAM-MD5 authentication (defined in RFC 2095). However, we have no plans to implement data-stream encryption for the connection.[1]

To use IMAIL, you must create an Edwin init file, called `~/.edwin` on unix machines or `edwin.ini` on Windows machines. This file contains arbitrary Scheme expressions that are evaluated in the Edwin environment when Edwin is started. In addition to any other customizations you put in this file, you must include the following expression:

```
(load-option 'imail)
```

Next, you must tell Edwin where to find your IMAP server, by setting some variables; the expression to do this must follow the call to `load-option`. Here is an example:

```
(load-option 'imail)
(set-variable! imail-default-imap-server "imap.foo.org")
```

Note that this is syntactically similar to Scheme's `set!` special form, but that it modifies the value of an Edwin editor variable rather than a Scheme variable. There are several other variables that control how IMAIL connects to the server. See Section 4.3 [Multiple Folders], page 9, for a complete list. By default, IMAIL tries to connect to 'localhost' using port 143, and to log in using the user name that you are logged in as. This is the right default if you are using stunnel on the client.

After you are finished creating the init file, you can either restart Edwin, or you can load the file using *M-x load-file*. At this point, you are ready to run IMAIL. To start IMAIL and read the mail in the 'inbox' folder on your IMAP server, type *M-x imail*.

---

[1] Here at MIT, we connect to our server using `stunnel` to provide end-to-end encryption. This provides connection security without the need to integrate the encryption into the client or the server.

3 Concepts

To use IMAIL effectively, it is helpful to know the terminology and understand the concepts underlying IMAIL's design. Here we will introduce you to messages, folders, URLs, and server connections.

3.1 Messages

A *message*, or *email message*, is the basic unit of electronic mail. The format of a message is defined by RFC 822. Nearly all email messages are transmitted over the internet, which means that the contents of such messages are further constrained by the SMTP protocol that is used for internet message transmission, as defined in RFC 821.

In brief, the primary constraints on an email message is that it may contain only printable US-ASCII characters, and that lines of text in the message may not exceed 1000 characters, *including* the carriage-return/linefeed pair at the end of each line. These constraints are fairly strict, and do not permit messages to contain text in languages other than English, or to contain non-textual data such as images. The *Multipurpose Internet Mail Extensions* (MIME, RFC 2045) provide a way to encode other kinds of text and data so that they can be carried in an email message. Most modern email software supports the MIME standard; one notable exception is Emacs Rmail.

3.2 Folders

Another important concept is a means for grouping messages together. All email software provides some means for doing this, and IMAIL is no exception. IMAIL provides objects called *folders*. A folder is just an object that holds an arbitrary number of email messages. Messages can be added to a folder, deleted from a folder, and moved or copied from one folder to another.

In IMAIL, the concept of the folder is used to embrace different grouping mechanisms. This is because IMAIL provides a uniform means for accessing different kinds of email systems. In particular, IMAIL supports access to Emacs Rmail files (also known as BABYL files, for historical reasons), to unix mailbox files (sometimes called *mbox* files), and to IMAP mailboxes. Each of these grouping mechanisms, although implemented very differently, is viewed as a folder by IMAIL. With some exceptions, each of these different *types* of folder are treated exactly the same by IMAIL. Finally, because IMAIL is extensible, other types of folders may be supported in the future.

3.3 Containers

Folders can themselves be grouped together inside objects called *containers*. A container is an object that holds folders, much in the same way that a folder holds messages. As for folders, the concept of a container is a generalization of the different kinds of mechanisms used by the underlying mail technology. So, for example, the container of a file folder is the directory holding that file.

IMAP containers are a little different: each IMAP mailbox is capable of holding other mailboxes. What this means is that, from IMAIL's point of view, an IMAP mailbox is both a folder object and a container object at the same time. However, when IMAIL views an

IMAP mailbox as a container, it is treated differently than when it is viewed as a folder, and consequently different notation is used in each case.

3.4 URLs

In email software like Rmail, where mail is stored in files, filenames are used to refer to groups of messages. Since IMAIL folders often aren't files, it is necessary to use a more general kind of reference for folders. To this end, IMAIL uses *Uniform Resource Locators* (URLs) to refer to folders.[1] IMAIL currently supports two kinds of URLs: IMAP URLs and file URLs.

3.4.1 IMAP URLs

The first kind of URL is an IMAP URL,[2] which looks like this:

```
imap://uname@hostname:port/mailbox
```

In this syntax, the parts '`uname@`' and '`:port`' are optional. *Hostname* is the internet host name or IP address of the IMAP server. *Uname* is the user name that identifies the account to be accessed on the server; this defaults to your user name. *Port* is the server's IP port; this defaults to `143` and is normally not specified.

Mailbox specifies the IMAP mailbox (or folder, in IMAIL's terminology) that is being referred to. Since most IMAP servers support hierarchical mailboxes, *mailbox* is a structured component indicating the location of the folder in the hierarchy, much like filenames or HTTP URLs. Here are some examples of IMAP URLs showing different mailbox paths:

```
imap://localhost/inbox
imap://localhost/inbox/sysadmin
imap://localhost/inbox/sysadmin/equipment
```

Here you see several interesting properties of IMAP mailboxes. The first URL refers to the primary IMAP mailbox for this account, called the *inbox*. All IMAP servers must support this mailbox, which is always called '`inbox`'; the name is not case sensitive and may be typed in any combination of upper or lower case letters. However, case sensitivity for names other than '`inbox`' is undefined by IMAP, so IMAIL treats all other names as if they were case sensitive.

The second and third URLs show how hierarchically-nested mailboxes are referred to: by writing the components of the path, separated by slashes. Note that IMAP does not require particular path-separator characters for hierarchical names, and in fact different IMAP servers use different separators. However, IMAIL *always* uses the forward-slash character as a separator, and translates to the server's character as needed.[3]

Another thing to note about these examples is that IMAP, unlike most file systems, allows a folder to contain messages *and* to have subfolders. This includes the '`inbox`' folder, as shown here. At least one server (Cyrus) puts *all* subfolders for a user account under '`inbox`', but this is not required by IMAP and is not generally true.

---

[1] URLs are defined in RFC 1738 and RFC 2396.

[2] The syntax for IMAP URLs is defined by RFC 2192, except that IMAIL uses only a subset of the defined syntax.

[3] This is in opposition to RFC 2192, which specifies use of the server-specific separator. RFC 2396 and RFC 2718 provide compelling arguments against this design.

3.4.2 File URLs

There is one other URL type supported by IMAIL: file URLs. This uses the 'file:' URL syntax,[4] as follows:

```
file://hostname/pathname
```

Here *hostname* refers to the host on which the file (folder) resides. Since IMAIL supports only files on the local file system, *hostname* must be 'localhost'; it may also be omitted, as in

```
file:///pathname
```

IMAIL also supports a non-standard abbreviation:

```
file:/pathname
```

As specified by the URL standard, *pathname* is a slash-separated sequence of path components, where unusual characters appearing in the components, such as the space character, are specially encoded. However, IMAIL will accept nearly any character in a component, and encode it if required; with few exceptions you can type any pathname without encoding. IMAIL always displays URLs with proper encoding.

In practice, this means that most unix filenames are written verbatim, with exceptions for special characters, and with the leading slash omitted. However, DOS-style filenames, as used by Windows, must be specially rewritten to conform to this style.

The rewriting rules for DOS file URLs are not specified by the standard, so consequently IMAIL defines its own rules for this encoding, as follows. A DOS filename is encoded by replacing all of the backslash characters with forward-slash characters, and by encoding unusual characters in the path components. Finally, the drive letter is prefixed to the path with an additional forward-slash separator. So for, example, the filename

```
C:\My Documents\Mail\My Mail.rmail
```

becomes the URL

```
file://localhost/C:/My%20Documents/Mail/My%20Mail.rmail
```

3.4.3 Container URLs

IMAIL also uses URLs to refer to containers. The notation used for a container is what you would expect: take a folder URL and drop everything after the rightmost slash. For example, the folder URL

```
imap://localhost/inbox/sysadmin/equipment
```

has the corresponding container URL

```
imap://localhost/inbox/sysadmin/
```

Note that this is different from the URL for the sysadmin folder:

```
imap://localhost/inbox/sysadmin
```

---

[4] File URLs are defined in RFC 1738.

3.5 Server Connections

Unlike a file folder, in which the folder's contents are always available, access to an IMAP folder requires an active network connection to the IMAP server. This adds an additional layer of complexity to the mail-reading process, which is reflected in the *connection state* of an IMAP folder.

An IMAP folder can be in one of two states: *online*, meaning that there is an established network connection between IMAIL and the IMAP server, and *offline* when there is not. IMAIL is, at present, a very simple IMAP mail reader: it must be online to read and manipulate mail messages. Mail readers that have this property are said to operate in *online mode*.[5] Do not confuse the online *state* with online *mode*. When we refer to online or offline in this document, it always means the corresponding *state*.

When an IMAP folder is selected in an IMAIL buffer, the modeline for that buffer shows either '`online`' or '`offline`' to indicate the folder's connection state. Normally, an IMAP folder goes online when it is first selected, and stays online indefinitely until it is explicitly disconnected.[6] Commands that break the connection are explicitly pointed out in their descriptions below; most other commands will force an IMAP folder into the online state if it is offline.

---

[5] IMAP also supports two other modes of operation, called *offline mode* and *disconnected mode*; at present IMAIL can not operate in these alternate modes.

[6] Although IMAP servers are allowed to disconnect mail readers that are inactive for long periods of time, IMAIL silently keeps the connection open by periodically transmitting commands to the server.

4 Commands

IMAIL provides a rich set of commands for manipulating messages. Like Rmail, most of these commands are bound to letter keys.

The most important command is `M-x imail`, which is used to start IMAIL. With no arguments, `M-x imail` reads the primary folder, selects the first unseen message in the folder, then selects the folder's buffer. If the primary folder is an IMAP folder, `M-x imail` will connect to the server and check for new mail. If `M-x imail` is given a prefix argument, it will prompt for the URL of a folder rather than reading the primary folder.

The IMAIL message buffer is put in `IMAIL` mode, a special mode in which most letter commands are defined to have special meanings. Where possible, the letters chosen for these commands are the same as those for the corresponding Rmail commands. The command keys specified in this chapter are for `IMAIL` mode, unless otherwise specified.

4.1 Navigation

The most basic thing to do with a message is to read it. The way to do this in IMAIL is to *select* the message. The usual practice is to move sequentially through the folder, since this is the order of receipt of messages. When you enter IMAIL, you are positioned at the first message that you have not yet seen (that is, the first one that has the 'unseen' flag; see Section 4.5 [Flags], page 17). Move forward to see the other new messages; move backward to reexamine old messages.

n   Move to the next nondeleted message, skipping any intervening deleted messages (`imail-next-undeleted-message`).

p   Move to the previous nondeleted message (`imail-previous-undeleted-message`).

M-n   Move to the next message, including deleted messages (`imail-next-message`).

M-p   Move to the previous message, including deleted messages (`imail-previous-message`).

j   Move to the first message. With argument *n*, move to message number *n* (`imail-select-message`).

>   Move to the last message (`imail-last-message`).

<   Move to the first message (`imail-first-message`).

M-u   Move to the first unseen message (`imail-first-unseen-message`).

M-s string RET
             Move to the next message containing a match for *string* (`imail-search`).

M-- M-s string RET
             Move to the previous message containing a match for *string*.

C-c C-n   Move to the next message with the same subject (`imail-next-same-subject`).

C-c C-p   Move to the previous message with the same subject (`imail-previous-same-subject`).

n and *p* are the usual way of moving among messages in IMAIL. They move through the messages sequentially, but skip over deleted messages, which is usually what you want to do. Their command definitions are named `imail-next-undeleted-message` and `imail-previous-undeleted-message`. If you do not want to skip deleted messages—for example, if you want to move to a message to undelete it—use the variants *M-n* and *M-p* (`imail-next-message` and `imail-previous-message`). A numeric argument to any of these commands serves as a repeat count.

In IMAIL, you can specify a numeric argument by typing just the digits. You don't need to type *C-u* first.

The *M-s* (`imail-search`) command is IMAIL's version of search. The usual incremental search command *C-s* works in IMAIL, but it searches only within the current message. The purpose of *M-s* is to search for another message. It reads a string nonincrementally, then searches starting at the beginning of the following message for a match. It then selects that message. If *string* is empty, *M-s* reuses the string used the previous time.

To search backward in the folder for another message, give *M-s* a negative argument. In IMAIL you can do this with `- M-s`.

It is also possible to search for a message based on flags. See Section 4.5 [Flags], page 17.

To find the next message with the same subject as the current message, use *C-c C-n* (`imail-next-same-subject`). This is useful for following the thread of an email conversation. *C-c C-p* (`imail-previous-same-subject`) finds the previous message with the same subject.

To move to a message specified by absolute message number, use *j* (`imail-select-message`) with the message number as argument. With no argument, *j* selects the first message. *<* (`imail-first-message`) also selects the first message. *>* (`imail-last-message`) selects the last message. *M-u* selects the first unseen message (`imail-first-unseen-message`).

4.2 Deleting Messages

When you no longer need to keep a message, you can *delete* it. This flags it as ignorable, and some IMAIL commands pretend it is no longer present; but it still has its place in the IMAIL folder, and still has its message number.

Expunging the IMAIL folder actually removes the deleted messages. The remaining messages are renumbered consecutively. Expunging is the only action that changes the message number of any message.

d   Delete the current message, and move to the next nondeleted message (`imail-delete-forward`).

C-d   Delete the current message, and move to the previous nondeleted message (`imail-delete-backward`).

u   Undelete the current message, or move back to a deleted message and undelete it (`imail-undelete-previous-message`).

x   Expunge the IMAIL folder (`imail-expunge`).

There are two IMAIL commands for deleting messages. Both delete the current message and select another message. *d* (`imail-delete-forward`) moves to the following message,

skipping messages already deleted, while `C-d` (`imail-delete-backward`) moves to the previous nondeleted message. If there is no nondeleted message to move to in the specified direction, the message that was just deleted remains current. A numeric argument to either command reverses the direction of motion after deletion.

To make all the deleted messages finally vanish from the IMAIL folder, type `x` (`imail-expunge`). Until you do this, you can still *undelete* the deleted messages. The undeletion command, `u` (`imail-undelete-previous-message`), is designed to cancel the effect of a `d` command in most cases. It undeletes the current message if the current message is deleted. Otherwise it moves backward to previous messages until a deleted message is found, and undeletes that message.

Because `imail-expunge` irreversibly deletes mail, IMAIL normally requires confirmation before it performs the expunge. This confirmation is controlled by the value of the variable `imail-expunge-confirmation`, which is a list of symbols. There are two independent behaviors controlled by this: whether to prompt, and whether to show the messages being expunged. If the list contains the symbol `verbose` (the default), the user is prompted for a yes-or-no style confirmation; if the list contains the symbol `brief`, the user is prompted for a y-or-n style confirmation; if neither of these symbols is present, no confirmation is done. If the list contains the symbol `show-messages`, a window is popped up showing the messages to be expunged; otherwise the list is not shown.

You can usually undo a `d` with a `u` because the `u` moves back to and undeletes the message that the `d` deleted. But this does not work when the `d` skips a few already-deleted messages that follow the message being deleted; then the `u` command undeletes the last of the messages that were skipped. There is no clean way to avoid this problem. However, by repeating the `u` command, you can eventually get back to the message that you intend to undelete. You can also select a particular deleted message with the `M-p` command, then type `u` to undelete it.

A deleted message has the '`deleted`' flag, and as a result '`deleted`' appears in the mode line when the current message is deleted. In fact, deleting or undeleting a message is nothing more than adding or removing this flag. See .

4.3 Multiple Folders

IMAIL operates by default on your *primary folder*, which is the folder named '`inbox`' on your IMAP server. Your incoming mail is placed in that folder by your system's mail-delivery software. Whenever it has an open connection to the server, IMAIL notices new mail and brings it to your attention by modifying the Edwin mode line.

You can specify a different folder to be your primary folder by modifying one or more of IMAIL's variables. The simplest way to do this is to change the variable `imail-primary-folder` to contain the URL of the folder that you wish to be your primary folder. Normally `imail-primary-folder` is `#f`, in which case the primary folder has the form

```
imap://user-id@server/mailbox
```

where *user-id* is the value of the variable `imail-default-user-id`, *server* is the value of `imail-default-imap-server`, and *mailbox* is the value of `imail-default-imap-mailbox`. `imail-default-user-id` may be `#f` meaning to use the value of '(`current-user-name`)'.

In addition to the primary folder, you can also have other folders and edit them with IMAIL. You can move messages into them with explicit IMAIL commands.[1]

One major difference between a file-based mail reader like Rmail and an IMAP mail reader like IMAIL is that file-based mail readers do not need to provide commands to manipulate mail files (as opposed to mail messages). This is because ordinary file-system commands already provide the ability to copy, delete, and rename such files. This isn't the case for IMAP mail readers. Consequently IMAIL provides a basic set of commands for manipulating folders, as well as a Dired-like folder browser.

4.3.1 Simple Folder Commands

Within a folder's buffer, IMAIL provides a number of simple commands that can be used to interact with other folders.

`i URL RET`   Read the folder named *URL* and run IMAIL on it (`imail-input`).

`g`   Get new mail for the current folder (`imail-get-new-mail`).

`C-u g URL RET`
              Read the folder named *URL* and append all of its messages to the current folder (`imail-input-from-folder`).

`o URL RET`   Copy the current message into the folder named *URL* (`imail-output`).

`M-o filename RET`
              Append the current message, in human-readable form, to the file named *filename* (`imail-file-message`).

`C URL1 RET URL2 RET`
              Copy the folder named *URL1* to *URL2* (`imail-copy-folder`).

`D URL RET`   Delete the folder named *URL* (`imail-delete-folder`).

`R URL1 RET URL2 RET`
              Rename the folder named *URL1* to be *URL2* (`imail-rename-folder`).

`+ URL RET`   Create a folder named *URL* (`imail-create-folder`).

To run IMAIL on a folder other than your primary folder, you may use the `i` (`imail-input`) command in IMAIL. This visits the folder in IMAIL mode. You can use `M-x imail-input` even when not in IMAIL.

The `g` (`imail-get-new-mail`) command gets new mail for the current IMAIL folder, and if there is new mail, moves to the first unseen message. This command works only on IMAP folders; it does nothing on file-based folders. Normally this command isn't needed since IMAIL periodically checks for new mail in all IMAP folders, but it is occasionally useful to force IMAIL to get new mail immediately rather than waiting for the next periodic mail check. The command `M-x imail` has the same effect as `imail-get-new-mail` if the primary folder is already open in a buffer.

---

[1] While Emacs Rmail additionally supports the ability to retrieve mail from "system inboxes" on your local computer (usually `/var/spool/mail/USER` on unix systems), IMAIL does not. IMAIL only supports incoming mail when it is delivered to an IMAP server. This Rmail feature can easily be implemented if desired, but there has been no call for it.

IMAIL normally checks for new mail in IMAP folders according to the value of the variable `imail-update-interval`. This variable specifies the time between checks in seconds. It may also be set to `#f`, which disables automatic mail checking. When IMAIL detects new mail in the primary folder, it normally modifies the mode line of all buffers to contain the string '`[New Mail]`'. This can be disabled by setting the variable `imail-global-mail-notification` to `#f`.

To copy messages from another folder into the current folder, give the *g* key a numeric argument, as in *C-u g*. This runs the command `imail-input-from-folder`, which reads a URL and copies all the messages from the specified folder into the current one. The messages are appended to the current folder, in the same order that they appear in the specified folder.

The *o* (`imail-output`) command copies the current message into a folder that you specify as a URL. The folder initially defaults to the current folder, unless you have set the variable `imail-output-default` to a different default; after the first message is output, the default folder becomes the one to which you last output a message. If the target folder doesn't exist, it is created first; in any case, the copied message is appended to the end of the folder. The current message is flagged as '`filed`'. If the variable `imail-delete-after-output` is true, the message is also marked as deleted.

The *M-o* (`imail-file-message`) command appends the current message to a file that you specify. The message is written to the file in the same format in which it appears in the IMAIL buffer, preceded by a separator line if there is other text in the file. This simplifies the task of collecting messages into a plain text file for future reference. Note that attachments appear in the output only as markers; their contents are not included in the file.

The *C* (`imail-copy-folder`) command copies an entire folder from one place to another. You specify two URLs, the source and the target, and all of the messages from the source folder are copied verbatim to the target folder. The source folder is not changed. The target folder is created if it doesn't exist. If the target folder does exist, the source folder's messages are appended to it.

Note that all of the commands that copy messages between folders will work whether the folders are the same type or not. In particular, messages in IMAP folders can be copied to file folders, and vice versa. You can copy messages between two file folders in different formats, or between two different IMAP servers. IMAIL doesn't care; it translates as needed.

The *D* (`imail-delete-folder`) command deletes a specified folder. All of the messages in the folder, and the folder itself, are deleted. You will be prompted to confirm before any deletion is done.

The *R* (`imail-rename-folder`) command renames a specified folder. You are prompted for two URLs, the old name and the new one. At present, this command only works in limited circumstances, specifically, when moving a folder from one place to another on a single IMAP server, or when moving a file folder from one place to another within the same file system. The rename operation fails if the new name is already in use.

The *+* (`imail-create-folder`) command creates a new, empty folder. It prompts for a URL, and signals an error if the name is already in use. This command is rarely used since the message-copying commands automatically create folders as needed.

4.3.2 The Folder Browser

In addition to the simple commands just described, IMAIL also provides a Dired-like browser for viewing and manipulating folders. The browser is generic, meaning that it will view collections of both IMAP folders and file folders, although it works better and is more useful in conjunction with IMAP folders.

The `imail-browser-view-container` command is used to enter the folder browser. In an IMAIL folder buffer, this command is bound to the `^` key, and will bring up a folder browser that is viewing the container of the current folder. With a prefix argument, you will be asked for the URL of a container to browse.

An IMAIL browser buffer is arranged so that each line in the browser represents a folder or a container (or both, in the case of IMAP containers). Here is an example:

```
imap://localhost/inbox/
----------------------
  + debian/
  + family
  + gnu/
  + ham/
  + hp-laptops/
  + linux/
  + misc/
  + mit/
  + music/
  + purchases
  + scheme/
  + software/
  + sysadmin
  + vendors/
  + vlsi
```

There are several interesting features of this buffer. The first two lines of the buffer are a title, telling you the URL of the container that this buffer is browsing. Each of the remaining lines shows the name of a folder (or container) that is inside the container. You can perform various operations on one of these folders by moving point to the folder's line and invoking commands.

Each line uses special characters to give you cues about the object being described. If the object is a container, there is a '+' character at the beginning of the line. Because our example is showing an IMAP container, and virtually all IMAP folders are also containers, every line in the example starts with '+'. Additionally, if the object is *only* a container, then the object's name ends in the character '/'; if the object is only a folder, or if it is both a folder and a container, then there is no trailing '/'.

The following commands are available in an IMAIL browser buffer.

f   View the folder on the current line in an IMAIL buffer (`imail-browser-view-selected-folder`).

t   If the object on the current line is a container, toggle whether its contents are shown (`imail-browser-toggle-container`).

`c`   Browse the container on the current line (`imail-browser-view-selected-container`).

`^`   Browse the container of the this buffer's container (`imail-browser-view-container`).

`d`   Mark the object on the current line for subsequent deletion (`imail-browser-flag-folder-deletion`).

`m`   Mark the object on the current line for subsequent operations (`imail-browser-mark`).

`u`   Remove any mark from the current line and move to the next line (`imail-browser-unmark`).

`DEL`   Move to the previous line and remove any mark there (`imail-browser-unmark-backward`).

`M-DEL char`
               Remove marks from all folders (`imail-browser-unmark-all-folders`).

`C URL RET`   Copy the folder on the current line to the folder named *URL*, creating it if needed (`imail-browser-do-copy`).

`R URL RET`   Rename the object on the current line to be *URL* (`imail-browser-do-rename`).

`D`   Delete the object on the current line (`imail-browser-do-delete`). Prompts for confirmation before performing the deletion.

`x`   Delete all objects that have been marked for deletion (`imail-browser-do-flagged-delete`). Prompts for confirmation before any deletion is performed.

`+ URL RET`   Create a folder named *URL* (`imail-create-folder`).

`g`   Recompute the buffer's contents by querying the server or file system (`imail-browser-revert`).

`q`   Kill the current buffer (`imail-browser-quit`).

If point is on a line describing a folder, use the `f` (`imail-browser-view-selected-folder`) command to view the contents of that folder. This selects an IMAIL folder buffer for that folder.

If point is on a line describing a container, use the `t` (`imail-browser-toggle-container`) command to show the contents of the container in the current buffer. This causes the '`+`' on this line to change to a '`-`', and new lines describing the contents are inserted into the buffer following the current line. The new lines are slightly indented to indicate the container relationship. For example:

```
imap://localhost/inbox/
-----------------------
  - debian/
        + bugs
        + maintainer
        + misc
  + family
  + gnu/
  + ham/
  + hp-laptops/
  + linux/
  + misc/
  + mit/
  + music/
  + purchases
  + scheme/
  + software/
  + sysadmin
  + vendors/
  + vlsi
```

To hide the container's content lines, use the `t` command again. Another way to open and close containers is to click the left mouse button on the '+' or '-' character for the container (`imail-browser-mouse-toggle-container`).

If you would rather browse a container in a separate buffer, use the `c` (`imail-browser-view-selected-container`) command. To browse the container of this buffer's container, use the `^` (`imail-browser-view-container`) command.

Besides simple browsing capabilities, the IMAIL folder browser also provides the ability to modify folders and containers, by copying, renaming, and deleting them. The commands to do this normally operate on the object on the current line. However, you can *mark* one or more lines, and subsequently perform an operation on all of them at once.

There are several marking and unmarking commands. The `m` (`imail-browser-mark`) command marks the current line and moves down to the next line. The mark is visible as an asterisk at the beginning of the line. A numeric argument serves as a repeat count. The `d` (`imail-browser-flag-folder-deletion`) is just like `m`, except that it marks lines with 'D'. 'D' marks are used to flag objects for deletion, while '*' marks are used for everything else.

To unmark a line, use the `u` (`imail-browser-unmark`) command. This removes any mark from the current line and moves to the next line. Like `m`, a numeric argument serves as a repeat count. The DEL (`imail-browser-unmark-backward`) command moves upward, removing flags; it is like `u` with argument `-1`. Finally, the `M-DEL` (`imail-browser-unmark-all-folders`) prompts for a character and unmarks *all* lines marked with that character; specifying RET as the character removes all marks.

The next three commands perform the copy, rename, and delete operations. These commands all operate on one or more folders, which you specify either by marking them, or by moving point to the corresponding lines. The folders to be operated on are specified

as follows. If the command is given a numeric argument N, then the next N folders are specified. Otherwise, any folders marked with an asterisk are specified. If there is no argument and no marked folders, then the folder on the current line is specified.

The `C` (`imail-browser-do-copy`) command copies one or more folders. If one folder is specified, the command prompts for the URL of another folder, and appends the messages of the first folder to the end of the second folder. The second folder is created if necessary. If more than one folder is specified, the command prompts for the URL of an existing container, and copies the source folders into the target container with the same names.

The `R` (`imail-browser-do-rename`) command renames one or more folders. If one folder is specified, the command prompts for a URL, and changes the name of the folder to the URL. If more than one folder is specified, the command prompts for the URL of an existing container, and moves the folders into the container. Note that in both cases, it is an error if there is already a folder with the new name. Furthermore, this command only works in limited circumstances, specifically, when moving a folder from one place to another on a single IMAP server, or when moving a file folder from one place to another within the same file system.

The `D` (`imail-browser-do-delete`) command deletes one or more folders. The command prompts for confirmation before any folders are deleted. The `x` (`imail-browser-do-flagged-delete`) command is similar, except that the folders it deletes are those that have been marked with 'D'. (The `x` command is mostly provided for compatibility with Dired.)

The `+` (`imail-create-folder`) command creates a new, empty folder. It prompts for a URL, and signals an error if the name is already in use. This command is rarely used since the message-copying commands automatically create folders as needed.

The `g` (`imail-browser-revert`) command re-reads the contents of the browser buffer's container and uses that information to regenerate the buffer's contents. Any marks that you have placed in the buffer are preserved.

The `q` (`imail-browser-quit`) command kills the current buffer. If you have marked some folders for later operation, the marks are discarded and the operations are not performed.

4.4 MIME Support

The *Multipurpose Internet Mail Extensions* (MIME) define a standard means for structuring mail messages. MIME permits a message to have multiple parts, each of which is called an *entity*. It also provides a way to associate type information with each entity. For example, an ordinary text message has type '`text/plain`', HTML has type '`text/html`', and a JPEG image has type '`image/jpeg`'. Additionally, MIME entities may be annotated to indicate whether they should be shown *in-line*, or whether they are *attachments* that should be shown only upon further user action.

IMAIL provides simple support for MIME messages. MIME attachments are shown in the IMAIL buffer by special abbreviations. You can write an attachment to a file. Multipart MIME structures are recognized and displayed in a clean format that suppresses unnecessary clutter. And MIME encodings such as *quoted-printable* and *base64* are automatically decoded prior to displaying the message or saving the attachment.

End-user formatting of MIME messages is a complex process, partly because these messages can be arbitrarily complex in their internal structure. IMAIL provides several variables that give you some control over the formatting prcess.

Many MIME messages have multiple parts; for example, a message with an attachment normally contains at least two parts: the message text and the attachment. IMAIL separates the different parts of a MIME message with specially-formatted lines. There are several styles of separator lines available, selected by changing the value of the variable `imail-mime-boundary-style`. The default value of `simple` means to use long lines of hyphen characters as the separator lines. A value of `sgml` means use long lines of hyphens that are wrapped with '`<!--`' and '`-->`', which makes them valid SGML comments. A value of `original` means to use the original MIME *boundaries*, which have certain useful syntactic properties but are not as visually distinctive.

MIME also specifies a particular kind of multipart message, of type '`multipart/alternative`', in which the parts are different representations of the same message. A typical example of this is a mailer that sends both plain text and HTML versions of the message text. Normally IMAIL shows only the simplest of these parts (which is almost always plain text) and suppresses the alternatives. However, if you set the variable `imail-mime-show-alternatives` to `#t`, IMAIL will show these alternative forms as attachments.

Another kind of multipart MIME message is the digest message, which has type '`multipart/digest`'. Digest messages are normally used by high-volume mailing lists to reduce the number of messages sent to the end user; instead the user receives one message containing all of the messages from that list in a particular time period, usually a day. IMAIL can present MIME digest messages in one of two formats. The default format is to show all of the component messages of the digest as attachments. This is particularly useful for large digests that you will only read a few messages from, since you can scan the digest contents for interesting messages without downloading all of the messages in the digest. In the alternative format, selected by setting `imail-mime-collapse-digest` to `#f`, the component messages of a digest are all shown inline.

As a general rule, any MIME entity that contains non-textual information is displayed as an attachment. Attachments are normally shown as specially-formatted abbreviations. Here is an example:

```
<imail-part
    name="foo.doc"
    type="application/msword"
    length="55499"
    />
```

This shows various things about the attachment, including its (optional) name, its MIME type, and the length of the attachment in bytes. (The length is computed on the encoded form of the attachment, and is generally slightly larger than the decoded length.)

IMAIL uses somewhat more complicated rules for deciding when a MIME entity is displayed in this abbreviated format, and when it is expanded in line. In general, all non-text entities are abbreviated. Additionally, if a text entity is given a *MIME disposition* of '`attachment`', if the character set of the entity is unknown, if the encoding type is unknown, or if the subtype is unknown, it is abbreviated.

Two variables control the abbreviation of text entities. `imail-known-mime-charsets` is a list of regular expressions that specify the known character sets; by default it specifies US-ASCII, the ISO 8859 character sets, and some random but commonly-seen Microsoft Windows character sets. The variable `imail-inline-mime-text-subtypes` contains a list of symbols, each of which is the name of a text subtype that should be shown in line. For example, if the symbol `html` is in this list, then MIME parts of type `text/html` are shown in-line. Text subtypes not appearing in this list are abbreviated as attachments.

Here are IMAIL's MIME-specific commands:

`C-o`   Save a MIME attachment to a file (`imail-save-attachment`).

`w`   Save an arbitrary MIME entity (message part) to a file (`imail-save-mime-entity`).

`C-c C-t C-e`
              Toggle a MIME entity between its formatted and raw forms (`imail-toggle-mime-entity`).

The primary MIME command is `C-o` (`imail-save-attachment`), which saves a single attachment to a file. If point is on an attachment, that is the attachment to be saved, otherwise IMAIL prompts for an attachment by name. If a prefix argument is specified, prompting is performed even if point is on an attachment. Once the attachment is determined, IMAIL prompts for the name of a file to save the attachment to. The filename is initialized from the name specified by the attachment, if any. The directory of the filename is initialized to the directory in which the last attachment was saved, or the user's home directory if no attachments have previously been saved.

If you want to save attachments to a specific directory, change the variable `imail-mime-attachment-directory` to contain the name of that directory.

The command `w` (`imail-save-mime-entity`) is similar to `imail-save-attachment` except that it will save any MIME entity, not just an attachment. For example, this allows you to save the message text. This command saves the entity that point is on; if point is not on any entity, an error is signalled. If the entity is encoded, e.g. with quoted-printable or base64 encoding, it is decoded before it is saved. If the entity is text, it is written to the file in text mode (relevant only under Windows); otherwise it is written in binary mode.

A simpler way to save a MIME entity is to point at the entity with the mouse and click the right button (`imail-mouse-save-mime-entity`). This works the same way as `imail-save-mime-entity` except that the entity is selected by the mouse instead of point.

The command `C-c C-t C-e` (`imail-toggle-mime-entity`) is similar to `imail-save-mime-entity`, except that instead of saving the entity to a file, it toggles whether the entity is shown in-line or in abbreviated form. A common situation in which this is useful is when the text of a message is in an unknown character set. In this case, IMAIL by default shows the text in abbreviated form; use `C-t` to expand it in place.

4.5 Flags

Each message can have various *flags* assigned to it as a means of classification. Each flag has a name; different names are different flags. Any given flag is either present or absent on a particular message. A few flag names have standard meanings and are given to messages automatically by IMAIL when appropriate. All other flags are assigned only by users.

`a flag RET`
> Assign the flag *flag* to the current message (`imail-add-flag`).

`k flag RET`
> Remove the flag *flag* from the current message (`imail-kill-flag`).

`C-M-n flags RET`
> Move to the next message that has one of the flags *flags* (`imail-next-flagged-message`).

`C-M-p flags RET`
> Move to the previous message that has one of the flags *flags* (`imail-previous-flagged-message`).

`C-M-l flags RET`
> Make a summary of all messages containing any of the flags *flags* (`imail-summary-by-flags`).

The `a` (`imail-add-flag`) and `k` (`imail-kill-flag`) commands allow you to assign or remove any flag on the current message.

Once you have given messages flags to classify them as you wish, there are two ways to use the flags: in moving and in summaries.

The command `C-M-n flags RET` (`imail-next-flagged-message`) moves to the next message that has one of the flags *flags*. The argument *flags* specifies one or more flag names, separated by commas. `C-M-p` (`imail-previous-flagged-message`) is similar, but moves backwards to previous messages. A numeric argument to either command serves as a repeat count.

The command `C-M-l flags RET` (`imail-summary-by-flags`) displays a summary containing only the messages that have at least one of a specified set of flags. The argument *flags* is one or more flag names, separated by commas. See Section 4.8 [Summaries], page 21, for information on summaries.

If the *flags* argument to `C-M-n`, `C-M-p` or `C-M-l` is empty, it means to use the last set of flags specified for any of these commands.

Some flags such as 'deleted' and 'filed' have built-in meanings and are assigned to or removed from messages automatically at appropriate times. Here is a list of built-in flags:

'seen'
> Means the message has been selected, implying that the user has seen it. Assigned to a message when it is selected by the user. When you start IMAIL, it initially shows the first message that lacks this flag.

'deleted'
> Means the message is deleted. Assigned by deletion commands and removed by undeletion commands (see Section 4.2 [Deleting Messages], page 8).

'filed'
> Means the message has been copied to another folder. Assigned by the message-copying commands (see Section 4.3 [Multiple Folders], page 9).

'answered'
> Means you have mailed an answer to the message. Assigned by the `r` command (`imail-reply`). See Section 4.6 [Sending Replies], page 19.

'`forwarded`'
> Means you have forwarded the message. Assigned by the `f` command (`imail-forward`). See Section 4.6 [Sending Replies], page 19.

'`resent`'   Means you have resent the message. Assigned by the command `C-u f` (`imail-resend`). See Section 4.6 [Sending Replies], page 19.

All other flags are assigned or removed only by the user, and have no standard meaning.

4.6 Sending Replies

IMAIL has several commands that use Mail mode to send outgoing mail. What this section documents are the special commands of IMAIL for entering Mail mode. Note that the usual keys for sending mail—`C-x m`, `C-x 4 m`, and `C-x 5 m`—are available in IMAIL mode and work just as they usually do.

`m`   Send a message (`imail-mail`).

`c`   Continue editing the already started outgoing message (`imail-continue`).

`r`   Send a reply to the current IMAIL message (`imail-reply`).

`f`   Forward the current message to other users (`imail-forward`).

`C-u f`   Resend the current message to other users (`imail-resend`).

The most common reason to send a message while in IMAIL is to reply to the message you are reading. To do this, type `r` (`imail-reply`). This displays the '`*mail*`' buffer in another window, much like `C-x 4 m`, but preinitializes the '`Subject`', '`To`', '`CC`' and '`In-reply-to`' header fields based on the message you are replying to. The '`To`' field starts out as the address of the person who sent the message you received, and the '`CC`' field starts out with all the other recipients of that message.

You can exclude certain recipients from being placed automatically in the '`CC`', using the variable `imail-dont-reply-to-names`. Its value should be a regular expression (as a string); any recipient that the regular expression matches is excluded from the '`CC`' field. The default value matches your own name, and any name starting with '`info-`' (the value of the variable `imail-default-dont-reply-to-names`). (Those names are excluded because there is a convention of using them for large mailing lists to broadcast announcements.)

To omit the '`CC`' field completely for a particular reply, enter the reply command with a numeric argument: `C-u r` or `1 r`.

By default, the '`Subject`' field of a reply is initialized to the contents of the '`Subject`' field of the message being replied to. However, if the variable `imail-reply-with-re` is set to `#t`, then the reply subject will be prefixed with '`Re:`'.

Once the '`*mail*`' buffer has been initialized, editing and sending the mail goes as usual. You can edit the presupplied header fields if they are not right for you. You can also use the commands of Mail mode, including `C-c C-y` which yanks in the message that you are replying to. You can switch to the IMAIL buffer, select a different message there, switch back, and yank the new current message.

Another frequent reason to send mail in IMAIL is to *forward* the current message to other users. `f` (`imail-forward`) makes this easy by preinitializing the '`*mail*`' buffer with the current message as a MIME attachment, and a subject designating a forwarded message.

All you have to do is fill in the recipients and send. When you forward a message, recipients get a message which is "from" you, and which has the original message in its contents.

By default, forwarded messages are sent as MIME attachments, which allows MIME-aware mail readers to recognize that the attachment is a mail message and to specially present it. However, this means that such forwarded messages appear more complex when viewed in mail readers that do not understand MIME. IMAIL deliberately minimizes the amount of encoding overhead used for MIME-forwarded messages, but some people prefer not to use MIME at all. For that reason, IMAIL allows you to turn off this feature, so that forwarded messages are included in the main body of the message (as Rmail does). To do this, set the variable `imail-forward-using-mime` to `#f`.

Normally, when IMAIL forwards a message, it sends only a few of the message's header fields. In particular, it sends only those header fields that you see when viewing the message in IMAIL. Sometimes it is desirable to send *all* of the message's header fields; IMAIL provides two ways to do this. First, if you want to send all of the header fields for a particular message, use `imail-forward` with a negative argument, like this: `- f`. Alternatively, you can set the variable `imail-forward-all-headers` to `#t`, which will cause *all* forwarded messages to retain all of their header fields.

Resending is an alternative similar to forwarding; the difference is that resending sends a message that is "from" the original sender, just as it reached you—with a few added header fields `Resent-from` and `Resent-to` to indicate that it came via you. To resend a message in IMAIL, use `C-u f`. (`f` runs `imail-forward`, which is programmed to invoke `imail-resend` if you provide a numeric argument.)

The `m` (`imail-mail`) command is used to start editing an outgoing message that is not a reply. It leaves the header fields empty. Its only difference from `C-x 4 m` is that it makes the IMAIL buffer accessible for `C-c C-y`, just as `r` does. Thus, `m` can be used to reply to or forward a message.

The `c` (`imail-continue`) command resumes editing the '`*mail*`' buffer, to finish editing an outgoing message you were already composing, or to alter a message you have sent.

4.7 Message Display

IMAIL provides several variables and commands to give you control over how messages are formatted in the message buffer.

By default, IMAIL automatically wraps long lines at the right margin. It uses *adaptive fill*[2] to do the wrapping, which means that common prefixes such as '`>`' and '`Chris>`' will be automatically copied down with the wrapped line. Generally, this wrapping makes messages easier to read. Specifically, it is important for messages sent by clients that use "soft" line breaks, because such clients expect the mail reader to wrap lines. However, if you'd rather not have IMAIL do this for you, you can disable wrapping by setting the variable `imail-auto-wrap` to `#f`.

Another feature of IMAIL is that it filters message headers, showing you only the most relevant ones. There are two variables that control how this is done, and a command that can override the filtering. The variable `imail-kept-headers` contains a list of regular expressions that are matched against message-header names (the name is everything to

---

[2] See the online help for the variable `adaptive-fill-regexp` for more information about adaptive fill.

the left of the colon, e.g. 'From' in 'From: cph'). If `imail-kept-headers` is a non-empty list, then only the headers matching those regular expressions are shown. Furthermore, the shown headers will be in the same order as the regular expressions.

If `imail-kept-headers` is an empty list, then all of the message headers are shown, except those matching the regular expression that is the value of the variable `imail-ignored-headers`. By default, the value of `imail-ignored-headers` contains some common uninteresting header names; this expression is identical to the default used by Rmail. Note that `imail-ignored-headers` is a single regular expression, while `imail-kept-headers` is a list of regular expressions. This is because `imail-ignored-headers` is meant to be an exact analog of the Rmail variable `rmail-ignored-headers`.

Regardless of how the message-header filtering is done, you can toggle between viewing the filtered headers and the unfiltered headers using the `t` command (`imail-toggle-header`). If filtered headers are shown, this command replaces them with unfiltered headers, and vice versa.

As you can see, IMAIL performs extensive transformation of a mail message before presenting it to you: MIME formatting, line wrapping, and header filtering. Sometimes, it's desirable to see the original message, exactly as it was received, without any formatting at all. The command `C-c C-t C-m` toggles the entire message between a formatted view and a raw view. This should be used with care, as a message with a large attachment might not fit in memory in its raw form.

4.8 Summaries

A *summary* is a buffer containing one line per message to give you an overview of the mail in an IMAIL folder. Each line shows the message number, the sender, the flags, and the subject. Almost all IMAIL commands are valid in the summary buffer also; these apply to the message described by the current line of the summary. Moving point in the summary buffer selects messages as you move to their summary lines.

A summary buffer applies to a single IMAIL folder only; if you are editing multiple IMAIL folders, each one can have its own summary buffer. The summary buffer name is made by appending '`-summary`' to the IMAIL buffer's name. Normally only one summary buffer is displayed at a time.

4.8.1 Making Summaries

Here are the commands to create a summary for the current IMAIL folder. Once the IMAIL folder has a summary buffer, changes in the IMAIL folder (such as deleting or expunging messages, and getting new mail) automatically update the summary.

`h`
`C-M-h`   Summarize all messages (`imail-summary`).

`l flags RET`
`C-M-l flags RET`
              Summarize messages that have one or more of the specified flags (`imail-summary-by-flags`).

`C-M-r rcpts RET`
              Summarize messages that have one or more of the specified recipients (`imail-summary-by-recipients`).

`C-M-t topic RET`
> Summarize messages that have a match for the specified regexp *topic* in their subjects (`imail-summary-by-topic`).

`C-M-s regexp RET`
> Summarize messages that have a match for the specified regexp anywhere in their header (`imail-summary-by-regexp`).

The `h` or `C-M-h` (`imail-summary`) command fills the summary buffer for the current IMAIL folder with a summary of all the messages in the folder. It then displays and selects the summary buffer in another window.

`C-M-l flags RET` (`imail-summary-by-flags`) makes a partial summary mentioning only the messages that have one or more of the flags *flags*. *flags* should contain flag names separated by commas.

`C-M-r rcpts RET` (`imail-summary-by-recipients`) makes a partial summary mentioning only the messages that have one or more of the recipients *rcpts*. *rcpts* should contain mailing addresses separated by commas.

`C-M-t topic RET` (`imail-summary-by-topic`) makes a partial summary mentioning only the messages whose subjects have a match for the regular expression *topic*.

`C-M-s regexp RET` (`imail-summary-by-regexp`) makes a partial summary mentioning only the messages whose headers contain a match for the regular expression *regexp*. This match includes all lines in the header, including for example the date and from lines.

Note that there is only one summary buffer for any IMAIL folder; making one kind of summary discards any previously made summary.

There are several variables that affect how summaries are displayed. `imail-summary-height` controls the height of the summary window. If it is an exact positive integer, the summary window is that many lines high. Alternatively, if it is a real number between 0 and 1 exclusive, the height of the summary window is computed by multiplying the number of lines in the message window by `imail-summary-height` and rounding to the nearest integer.

If the variable `imail-summary-fixed-layout` is set to `#t`, then an IMAIL message buffer and its associated summary buffer are always shown in a fixed window layout (provided the summary buffer exists). Selecting either buffer causes all other windows in the current editor frame to be deleted, then splits the frame into two windows showing the summary buffer and message buffer. Selecting any other buffer when this layout is displayed causes both windows to be deleted, and the other buffer displayed in a single window filling the editor frame. Basically, this causes the message and summary buffers to be treated as a unit most of the time. Currently, the default for `imail-summary-fixed-layout` is `#f`, but the feature has been very popular and the default may be changed to `#t` in a future release.

When fixed layout is not used, the variable `imail-summary-pop-up-message` provides a different kind of window splitting for message and summary buffers. If this is set to `#t` (the default), then selecting a new message in the summary buffer causes the message buffer to be popped up in a new window if it isn't already visible.

If the variable `imail-summary-highlight-message` is `#t`, the message currently selected in the message buffer is highlighted in the summary buffer. This aids navigation in the summary buffer and is thus the default. Set it to `#f` if you don't like the highlighting.

By default, the summary buffer has five columns: flags, message number, message length, subject, and author. If `imail-summary-show-date` is set to `#t`, a sixth column containing an abbreviated date appears between the message number and the subject.

The width of the subject column is specified by the value of `imail-summary-subject-width`; subject fields longer than this number of characters are truncated.

4.8.2 Editing in Summaries

You can use the IMAIL summary buffer to do almost anything you can do in the IMAIL buffer itself. In fact, once you have a summary buffer, there's no need to switch back to the IMAIL buffer.

You can select and display various messages in the IMAIL buffer, from the summary buffer, just by moving point in the summary buffer to different lines. It doesn't matter what Emacs command you use to move point; whichever line point is on at the end of the command, that message is selected in the IMAIL buffer.

Almost all IMAIL commands work in the summary buffer as well as in the IMAIL buffer. Thus, `d` in the summary buffer deletes the current message, `u` undeletes, and `x` expunges. `o` outputs the current message to a folder; `C-o` saves an attachment from the current message; `r` starts a reply to it. You can scroll the current message while remaining in the summary buffer using `SPC` and `DEL`.

The IMAIL commands to move between messages also work in the summary buffer, but with a twist: they move through the set of messages included in the summary. They also ensure the IMAIL buffer appears on the screen (unlike cursor motion commands, which update the contents of the IMAIL buffer but don't display it in a window unless it already appears). You can always display the message indicated by point using the `e` command (`imail-summary-select-message`).

When you are finished using the summary, type `C-x k RET` to delete the summary buffer's window.

4.9 Sorting Messages

IMAIL provides several commands that change the order of messages within a folder. These commands are closely modelled on those of Rmail. Note that these commands don't modify the folder itself; they only modify the way that the folder behaves. This means that if a folder buffer is deleted, the sort order associated with that buffer is deleted too.

`C-c C-s C-v`

> Sort messages of current IMAIL folder by arrival time (`imail-sort-by-arrival`). This is the default; this sort undoes any other sorting on the folder.

`C-c C-s C-d`

> Sort messages of current IMAIL folder by date (`imail-sort-by-date`).

`C-c C-s C-s`

> Sort messages of current IMAIL folder by subject (`imail-sort-by-subject`).

`C-c C-s C-a`

> Sort messages of current IMAIL folder by author's name (`imail-sort-by-author`).

`C-c C-s C-r`
          Sort messages of current IMAIL folder by recipient's names (`imail-sort-by-recipient`).

`C-c C-s C-c`
          Sort messages of current IMAIL folder by the name of the other correspondent (`imail-sort-by-correspondent`).

The IMAIL sort commands perform a *stable sort*: if there is no reason to prefer either one of two messages, their order remains unchanged. You can use this to sort by more than one criterion. For example, if you use `imail-sort-by-date` and then `imail-sort-by-author`, messages from the same author appear in order by date.

With a numeric argument, all these commands reverse the order of comparison. This means they sort messages from newest to oldest, from biggest to smallest, or in reverse alphabetical order.

4.10 Other Commands

This section documents a handful of commands and variables that don't fit into any of the other documentation categories.

`q`   Quit out of IMAIL (`imail-quit`). This closes all open IMAP connections and buries all IMAIL buffers. With prefix argument, only affects the current IMAIL buffer, leaving any other IMAIL buffers alone.

`M-d`   Disconnect from the IMAP server (`imail-disconnect`).

`b`   Bury the IMAIL buffer (`imail-bury`).

`s`   Save changes in the current folder to disk (`imail-save-folder`).

When you are finished reading mail in a folder, use the `q` command (`imail-quit`). This command *closes* the folder, then buries the buffer. Closing a folder has different effects, depending on the type of folder. Closing an IMAP folder causes IMAIL to disconnect from the IMAP server (go offline). Closing a file folder saves any changes out to the corresponding file. In both cases, internal data structures may be dropped, requiring them to be rebuilt, should the folder later be re-opened. In no case are any changes made to the folder's contents; in particular, deleted messages are *not* expunged.

On IMAP folders, the `q` command is equivalent to `M-d b`. On file folders, the `q` command is equivalent to `s b`.

The `M-d` command (`imail-disconnect`) disconnects IMAIL from the IMAP server (goes offline). This has no effect on file folders.

The `b` command (`imail-bury`) buries the selected IMAIL buffer. *Burying* a buffer means moving it to the bottom of the buffer list and selecting another buffer from the top of the list. This is similar to the command `bury-buffer`, except that any summary buffer associated with this buffer is also buried, and if a window was created to hold the summary buffer, it is deleted.

The `s` command (`imail-save-folder`) saves out any changes to the selected IMAIL folder. For file folders, this means writing the folder back out to its file. For IMAP folders,

this no effect. In no case are any changes made to the folder's contents; in particular, deleted messages are *not* expunged.

IMAIL normally caches IMAP message bodies in memory in order to increase performance. The variable `imail-body-cache-limit` gives you some control over how this caching is done. `imail-body-cache-limit` is normally set to a positive integer, meaning that any message body or in-line MIME entity whose size in bytes is less than this number is cached. However, caching can be entirely disabled by setting `imail-body-cache-limit` to `#f`, or made unconditional by setting it to `#t`.

Another thing that IMAIL caches is IMAP passwords. This is done so that you don't have to keep typing your password whenever you connect to a new IMAP folder. However, this is also a security risk, because the password is kept in Scheme's memory. The variable `imail-pass-phrase-retention-time` says how long passwords are cached, in minutes. Normally this is set to 30 minutes, but if you are paranoid you can set it to zero to disable password caching altogether. Scheme keeps track of the use of each password, and deletes its copy of the password when it has expired. Additionally, Scheme stores passwords in an obscured form, to prevent them being seen during casual browsing through memory structures, but this does *not* provide any protection against a deliberate attempt to find the password.

Appendix A GNU Free Documentation License

Version 1.2, November 2002

Copyright © 2000,2001,2002 Free Software Foundation, Inc.
51 Franklin St, 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 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 Warrany 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.

A.1 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 B Key Index

Appendix C Command Index

Appendix D Variable Index

Appendix E Concept Index