MM Electronic Mail
User Manual

Joseph Brennan

Columbia University Center for Computing Activities

Academic Computing Consulting Services

March 1990 Edition, revised December 1990

Some of the content of this manual is taken from earlier MM Manuals written by Sue Zayac, Lisa Covi and Lynn Jacobsen. The current editor and writer is Joseph Brennan, e-mail address brennan@cunixf.cc.columbia.edu. Comments and suggestions for improvements in future editions are welcome.

The Unix version of MM was written with CCMD at Columbia University by Chris Maio, Howie Kaye, Fuat Baran, and Melissa Metz. CCMD was also written at Columbia University, by Andrew Lowry and Howie Kaye.

The original MM for DEC20s (and its design) was written by a number of people, including Michael McMahon, Stuart McLure Cracraft, Ted Hess and Mark Crispin.

Table of Contents

Preface

The electronic mail system MM runs on CUCCA's unix-based computers. If you have an account on one of those machines, you are welcome to use MM to communicate with people on campus or around the world.

All faculty and staff of Columbia University are being offered free accounts with a limited amount of connect time and disk space. For information, call the Academic Computing Help Desk at 854-4854. The host computer is known as "Mail" or cunixf.

Students (and faculty using computers for instruction) have accounts on cunixa, cunixb or cunixd. There is a fee for students. To arrange for an account, contact the Business Office, 854-3555.

MM FROM YOUR OWN PC

Many MM users connect from their own PC's, on campus or off. That makes it convenient to check mail every day, or many times a day.

The booklet ***Preparing for Data Communications with the PBX*** gives details of how to connect through the ROLM phone system on the Morningside Campus. If you order the *data phone option* from Telecommunications (854-6254), you can simply run a cable between your PC's serial port and the connector built into the ROLM data phone. No modem is used.

From off campus, you can dial in to the ROLM PBX using a modem. The phone numbers are 854-1812 and 854-1824. They connect at 2400, 1200 or 300 baud.

Connections can also be made through some departmental computers on the Morningside Campus and at the Health Sciences Campus. Contact your local departmental administrator for details.

COMMUNICATIONS SOFTWARE

No matter how you connect, you need communications software to let your PC act as a terminal. Columbia supports software called *Kermit*, available for both MS-DOS PC's and Macintosh. Kermit can also transfer files between your PC and the MM host computer.

Macintosh and MS-DOS Kermit can be copied free at the Business Office, 102 Philosophy Hall; bring your own disk. At the Business Office, you can also purchase Kermit disks and manuals, and get free handouts. For more information on Kermit, contact Kermit Information at 854-3703.

LOGGING IN

Information on how to log in is contained in the handout ***Using MM***. You should get it when you get your account. Once you log in, you can start MM.

PUBLICATIONS

You can get all the publications mentioned here at the Academic Computing Business Office, 102 Philosophy Hall, 854-3555. The microcomputer labs at 215 International Affairs and 251 Mudd also have the handouts and reference copies of the manuals.

1. Introduction

* What Is MM?

MM is a powerful electronic mail system that allows you to send, read, edit, and manage messages quickly and easily.

MM has three different *modes*.

- **Top-level** is the default or initial mode you are in when you enter MM. At **top-level mode** you can perform general operations on your mail such as finding out how many messages are in your mailbox and who they are from. Also at **top-level**, you begin the procedures to *read* and *send* mail.
- In **read mode**, you give commands about the mail you are reading. You can for example delete or file it, or reply to it.
- In **send mode**, you give commands about the mail you are sending to other users.

Within each mode there are a series of commands that you may use to work with the messages you have received or to compose and send outgoing messages. A command simply consists of an English word, or several, typed at the *MM>*, *Read>*, or *Send>* prompt (the prompts at the three modes) followed by a carriage return.

Unlike other Unix programs, MM accepts a command in either upper or lower case. (Notice though that file names are case-sensitive.) MM will attempt to interpret each command: if understood, the command is executed; if not, MM will complain to you about the command error.

MM has a large list of commands that do many different things. You don't need to know them all to use MM effectively, but they make a flexible whole within which MM can operate.

* Learning MM

A good way to learn how to use MM and its commands is to look at this manual while you use MM. The second chapter is a series of examples demonstrating the use of the most common MM commands and should be helpful in getting you started. The other chapters offer a more detailed description of the commands used in the examples, as well as a complete list of all commands and options available at each mode.

If, after you have used MM extensively, you feel that something has been left out of the program, feel free to use the *bug* command, which lets you communicate with the maintainers of MM. We are willing to listen to reasonable suggestions and let you know whether anything can be done.

* Help

MM has extensive internal help. It is more detailed than this manual. If you are getting unexpected results or don't see quite how to use a command, see whether the online help explains it more fully.

There are three versions of help, as shown in the following examples:

Typing in a ***?*** at any MM prompt gives all the commands available at that point. The most important example is *help ?* (*help*, space, question-mark), which shows what help is available:

```
MM>help ?
 BASIC command, one of the following:
exit             help            headers         quit            read
review           send            suspend

 or MESSAGE-HANDLING command, one of the following:
answer           delete          forward         print           remail
reply            type            undelete

or MESSAGE-TAGGING command, one of the following:
flag             keyword         mark            unanswer        unflag
unkeyword        unmark

 or FILING command, one of the following:
copy             examine         expunge         get             move
restore-draft    sort            write

 or CUSTOMIZATION command, one of the following:
define           profile         save-init       set             show

 or INFORMATION command, one of the following:
check            count           daytime         finger          pwd
status           version         who

 or some OTHER command, one of the following:
backtrack        blank           browse          bug             bye
cd               continue        echo            edit            follow
jump             list            literal         next            previous
push             route           spell           take

  or "!" for shell escape

  or other topic, one of the following:
 addressing          basic               bitnet              ccmd
 command-history     command-line-edit   customization       directory
 filing              information         internet            message-handling
 message-sequence    message-tagging     other               shell
 signature-file      text-mode           top

  or other mode, one of the following:
 top-level-mode  read-mode       send-mode

  or confirm for a brief help message
```

At the *Send*> and *Read*> prompts, slightly different lists appear.

Typing in ***help*** followed by any command name then gives further information on each of the specific commands. For example, to get more information on the command *headers*, type ***help headers***:

```
MM>help headers

The HEADERS command is used to list "headline" summaries of messages
  in your current mail file.  Each summary line contains the sequence
  number, date, sender, subject, status and length of a message.

To use the HEADERS command, type "headers" followed by a message
  number or message-sequence.  The default is the current message.

For example:                    Displays summaries of:
  headers 2:4                     Messages numbered 2 through 4
  headers subj meeting            Messages with "meeting" in the subject line
  headers unanswered              Messages you haven’t replied to yet
  h from walter after 11/23/88    Messages from "walter" dated after
                                     Nov 23, 1988

The first example would give you the following:

    K  2) 23-Dec Walter Bourne   SAS graph stuff (1061 chars)
       3)  6-Feb Jeff Eldredge   Technical Notes (749 chars)
 FA    4) 21-Feb Bea Hamblett    sas article (277 chars)

For more help type "help" and one of these topics:
  message-sequence   basic
```

Typing ***headers ?*** gets a screen suggesting possible completions:

```
MM>headers ? message number
  or range of message numbers, n:m
  or range of message numbers, n-m
  or range of message numbers, n+m (m messages beginning with n)
  or "." to specify the current message
  or "*" to specify the last message
  or message sequence, one of the following:
 after              all                answered           before
 current            deleted            flagged            from
 inverse            keyword            last               longer
 new                on                 previous-sequence  recent
 seen               shorter            since              subject
 unflagged          unkeyword          unseen
  or "," and another message sequence
```

2. Sample Sessions and Examples

This section presents a series of sample sessions demonstrating the use of MM. These topics will be covered:

- Getting Started
- Sending a Message
- Reading Mail
- Replying to Messages
- Headers
- Managing Your Mail File: Delete and Undelete
- Reading Old Messages
- Forwarding Messages
- Copying To and From Regular (Non-Mail) Files
- File Transfer Using Kermit
- Using Emacs from within MM
- Additional Mail Files
- Defining Mailing Lists

In the following examples, what you type appears in **bold**. After most of the commands, hit the carriage return, marked *return* or *enter* on most keyboards, and shown here by the abbreviation **<*cr*>**. You do not have to type a space before the carriage return.

The *escape* key is indicated by the symbol ***[ESC]***; it may be marked *meta* or *alt* on some keyboards. Control Sequences are represented by **<*ctrl/*>** or ***CTRL-*** or the caret **^** followed by a letter. To type a Control letter, depress the control key while typing the letter (similar to pressing shift for upper case).

* Getting Started

MM can be started by simply typing ***mm*** to the Unix shell command interpreter. Most people at Columbia use the kornshell (ksh), which is identified by the $ prompt. So to start MM you would type:

```
$ mm <cr>
```

MM then checks to see whether you have new mail. If you do, it displays a line indicating what the message number is, the date it was received, who the message is from, the subject and the length of the message in characters.

```
$ mm <cr>
Columbia MM, version 0.90.0
Please report all bugs using MM's BUG command, or send mail to BUG-MM.
Suggestions are also welcome.
Reading /u/student/CC/lmc/mbox
19 messages read
N      20) 18-May Rob Cartolano       Meeting Canceled (475 chars)

[ H=headers R=read REV=review S=send Q=quit BYE ?=Hints HELP ]
MM>
```

In the example above, the user typed ***mm***. The next three lines show a greeting message. Then *Reading...* indicates that the old mail is being found in the file /u/student/CC/lmc/mbox. *19 messages read* indicates what was already stored in the mail file. The next line indicates that the user has one piece of new mail, from Rob Cartolano. Finally, there is a "hint" line, and the *MM>* prompt, which is what MM types to show that it is waiting to receive a command from you.

The "new mail" line is called a header. It is actually a combination of several header fields that you will become familiar with later on. What do we learn from it? The message is *N*ew (not read yet), it's message number *20* in the mail file, and it was sent on *18-May* by *Rob Cartolano*. Rob put the subject as *Meeting Canceled* and the message is *475 chars* long.

Usually, you read your new messages each time you start MM, but you don't have to. The headers of the new messages are shown so you can decide about reading them, based on who sent a message, what it's about, and how long it is.

* Sending a Message

Now that we are in MM, we would like to send a message. Sue is sending mail to Walter with a copy to Maurice. First she starts the MM program, then she types ***send***. MM prompts for the addressee, any carbon copies (cc:), and the topic of the message. After Sue provides this information, she types in the body of the message. When that is finished, she hits the ESC key.

```
$ mm
Columbia MM, version 0.90.0
Please report all bugs using MM's BUG command, or send mail to BUG-MM.
Suggestions are also welcome.
Reading /us/us/sue/mbox
8 messages read

[ H=headers R=read REV=review S=send Q=quit BYE ?=Hints HELP ]
MM>send
 To: walter
 cc: maurice
 Subject: SPSSX TNote Draft
 Message (End with CTRL-D or ESC
  Use CTRL-B to insert a file, CTRL-E to enter editor, CTRL-F to run text
  through a filter, CTRL-K to redisplay message, CTRL-L to clear screen and
  redisplay, CTRL-N to abort, CTRL-P to run a program and insert output.):

I think this is now ready for critical reading.  Do you
want to find readers or shall I?   /sue

[ D=display S=send TE=text ED=edit TY=type Q=quit ?=Hints HELP ]
Send>s
walter... Queued
maurice...Queued

[ H=headers R=read REV=review S=send Q=quit BYE ?=Hints HELP ]
MM>exit
No messages deleted.
[1] + Stopped (signal)          mm
$
```

In the above example, typing the ESC key terminated the message and left Sue in **send mode**, indicated by the *Send*> prompt. *Send mode* has many of its own commands that you can see by typing ***?***. To send the message off, she typed ***s*** to the *Send*> prompt. To exit MM she typed ***exit***. If she had decided NOT to send the message, she could have typed ***quit*** at the *Send* prompt.

KEEPING A COPY FOR YOURSELF

You may want to have a copy of a message he or she has sent. MM does not automatically keep a copy, but there are several ways of accomplishing it. One method is to *cc* yourself, and you will receive a copy of the message in your mailbox. A second method is to use the *fcc* command, which sends a copy of the message to a file that you name. In the example, both are done for the sake of demonstration.

```
$ mm
Columbia MM, version 0.90.0
Please report all bugs using MM's BUG command, or send mail to BUG-MM.
Suggestions are also welcome.
Reading /us/us/sue/mbox
8 messages read

[ H=headers R=read REV=review S=send Q=quit BYE ?=Hints HELP ]
MM>send
 To: walter
 cc: maurice, sue
 Subject: SPSSX TNote Draft
 Message (End with CTRL-D or ESC
  Use CTRL-B to insert a file, CTRL-E to enter editor, CTRL-F to run text
  through a filter, CTRL-K to redisplay message, CTRL-L to clear screen and
  redisplay, CTRL-N to abort, CTRL-P to run a program and insert output.):

I think this is now ready for critical reading.  Do you want to
find readers or shall I?   /sue

[ D=display S=send TE=text ED=edit TY=type Q=quit ?=Hints HELP ]
Send>fcc outmsg
Send>s
walter... Queued
maurice...Queued
sue...Queued
*outmsg...Sent

[ H=headers R=read REV=review S=send Q=quit BYE ?=Hints HELP ]
MM>exit
No messages deleted.
[1] + Stopped (signal)          mm
$
```

If you want to keep copies of everything you send, you can customize MM so that it does so, using one of the *set* commands described starting on page 38.

The customization commands *set default-cc-list* and *set default-fcc-list* can be used to put yourself on every cc list, or a certain file on every fcc list.

If you use *fcc*, either directly or through *set default-fcc-list*, note that it assumes the file goes into the current directory. This may cause problems if you change from your usual directory and then use MM. You have to start MM from the right directory, or specify the directory in the *fcc*. The customization command *set saved-messages-file* is similar to *set default-fcc-list*, but will always assume the file is in your mail-directory, and may therefore be preferable.

The file where the copies go is referred to as an additional mail file. To look at it, at the *MM>* prompt type ***get <filename>***, and you can then use all the MM commands with the file, like *headers*, *read*, and so on. To return to the main mail file, just type ***get***.

* Reading Mail

The mail sent by Sue in the previous example is now read by Walter.

```
$ mm
/usr/local/mm
N      6) 18-May Sue Zayac       SPSSX TNote Draft (251 chars)

[ H=headers R=read REV=review S=send Q=quit BYE ?=Hints HELP ]
MM>read
 Message 6 (251 chars)
Received: by cunixf.columbia.edu (5.54/5.10) id AA12719;
        Fri, 18 May 90 15:53:45 EDT
Date: Fri, 18 May 1990 15:53:43 EDT
From: Sue Zayac <sue@cunixf.columbia.edu>
To: walter
Cc: maurice
Subject: SPSSX TNote Draft
Message-Id: <CMM.0.90.580161223.sue@cunixf.columbia.edu>

I think this is now ready for proofreading.  Do you want to
find readers or shall I?      /sue
[Press any key to continue]

[ D=delete H=header R=reply TY=type PRI=print Q=quit ?=Hints HELP ]
Read>
```

After showing Walter the message, MM remained in **read mode**, indicated by the *Read>* prompt. **Read mode**, like **send mode**, has several of its own commands. Type a ***?*** at the *Read>* prompt to see them. If you press **<*cr*>** at the *Read>* prompt, the next unseen message is displayed. When there are no more, **<*cr*>** returns you to the top-level prompt, *MM>*.

* Replying to Messages

The *reply* command simplifies sending a message in reply to one you have received. In this example, Walter has just read Sue's message and sends her a reply.

```
Read>reply
 Message (End with CTRL-D or ESC
  Use CTRL-B to insert a file, CTRL-E to enter editor, CTRL-F to run text
  through a filter, CTRL-K to redisplay message, CTRL-L to clear screen and ]
  redisplay, CTRL-N to abort, CTRL-P to run a program and insert output.):

Have Lisa and Lynn read it. - Walter

[ D=display S=send TE=text ED=edit TY=type Q=quit ?=Hints HELP ]
Send>s
sue... Queued

[ D=delete H=header R=reply TY=type PRI=print Q=quit ?=Hints HELP ]
Read>
```

Reply is similar to *send*. *Reply* automatically fills in "To:" as the original sender, and copies the "Subject:" from the original message.

REPLY ALL and **REPLY INCLUDING**

Two keywords can be added to the reply command. Normally, the reply goes just to the sender of the original message, even though there may be other people who received the original message as part of its "To:" or "cc:" lists. If you command ***reply all***, the reply will be sent to all the people who got the original.

The second keyword, *including*, includes the original message in the reply. In the example above, for example, if Walter had delayed replying, Sue might have forgotten what Walter's reply was about. To guard against that, Walter could have typed ***reply including***, so Sue would receive a message containing her own message followed by Walter's response. The included text is set off by > marks.

Both keywords can be used together, ***reply all including***.

* Headers

Another useful command is *headers*. *Headers* will give you a headline summary of messages in your mail file. In this example, Walter uses the command *headers all* to get a list of all his mail messages. [Note: *all* here is a message sequence that refers to all messages contained in the current mail file. For a complete explanation of what message sequences are and which sequences are available, see page 24].

```
$ mm
/usr/local/mm

[ H=headers R=read REV=review S=send Q=quit BYE ?=Hints HELP ]
MM>headers all
       1) 17-May Margarita Suarez    LaserWriter B (317 chars
       2) 17-May Charlie C. Kim      sunos 4.0 (313 chars)
  A    3) 18-May Don Lanini          Emacs question (1036 chars)
       4) 18-May Robert C Lehman     Mets on Sunday (285 chars)
  A    5) 18-May Robert T. Cartolano Mac stuff (5258 chars)
R A    6) 18-May Sue Zayac           SPSSX TNote Draft (251 chars)

[ H=headers R=read REV=review S=send Q=quit BYE ?=Hints HELP ]
MM>
```

Messages that have been answered are marked with an "A" in the first column area. See the next page.

A useful variant of the *headers* command is *headers from <name>* (where *from <name>* is another example of a message sequence) to see only messages from certain people. For example:

```
MM>headers from su
       1) 17-May Margarita Suarez    LaserWriter B (317 chars)
R A    6) 18-May Sue Zayac           SPSSX TNote Draft (251 chars)

[ H=headers R=read REV=review S=send Q=quit BYE ?=Hints HELP ]
MM>
```

This is a string search, not a keyword search. Typing only part of the name ("su") will match any name with "su" in it.

Two very useful variants of the *headers* command are *headers subject <string>* and *headers text <string>*, which look for any messages with the string in their subject or text fields, respectively. Sometimes you remember getting a message about something, but not when or from whom. Choose a good topic word (or partial word) and one of these commands should help find the message. For example:

```
MM>headers subj su
       2) 17-May Charlie C. Kim      sunos 4.0 (313 chars)
       4) 18-May Robert C Lehman     Mets on Sunday (285 chars)

[ H=headers R=read REV=review S=send Q=quit BYE ?=Hints HELP ]
MM>
```

Again, note that this is a string search, not a keyword search. Think about this when choosing the "Subject" for your own mail. Be specific so your correspondents can do efficient *headers* searches. For example, don’t use something like "meeting"; use "Thursday Brown Bag Meeting".

Type ***help message-sequence*** at the *MM>* prompt to see all the variations of the *headers* command.

The letter codes at the left side of the *headers* display show the status of the message, as follows:

Message Header Abbreviations		
Letter	Name	Meaning
A	answered	You sent a *reply* or *answer* to the message.
D	deleted	You marked the message for deletion. It will be erased from your message file the next time you type *exit* or *expunge*.
F	flagged	You used the *flag* command to mark the message.
N	new	The message is both *recent* and *unseen* (see *R* and *U*).
K	keyword	You used the *keyword* command to mark a keyword on the message.
R	recent	The message has arrived as of this MM session.
U	unseen	The message has never been *type*d or *read.*
no letter		The message has already been seen.

* Managing Your Mail File: Delete and Undelete

You can get rid of messages you no longer need with the *delete* command:

```
MM>delete 1,2
   1:2

[ H=headers R=read REV=review S=send Q=quit BYE ?=Hints HELP ]
MM>headers all
   D   1) 17-May Margarita Suarez    LaserWriter B  (317 chars)
   D   2) 17-May Charlie C. Kim      sunos 4.0 (313 chars)
  A    3) 18-May Don Lanini          Emacs question (1036 chars)
       4) 18-May Robert C Lehman     Mets on Sunday (285 chars)
  A    5) 18-May Robert T. Cartolano Mac stuff (5258 chars)
R A    6) 18-May Sue Zayac           SPSSX TNote Draft (251 chars)

[ H=headers R=read REV=review S=send Q=quit BYE ?=Hints HELP ]
MM>
```

Messages that have been deleted are marked with a "D" in the first column area. Note that after messages are *delete*d, they are still there until you type ***exit*** and leave MM or until you type the command ***expunge***. This is useful if you should change your mind and want to *undelete* a *delete*d message:

```
MM>undel 2
 2

[ H=headers R=read REV=review S=send Q=quit BYE ?=Hints HELP ]
MM>headers all
   D   1) 17-May Margarita Suarez    LaserWriter B (317 chars)
       2) 17-May Charlie C. Kim      sunos 4.0 (313 chars)
  A    3) 18-May Don Lanini          Emacs question (1036 chars)
       4) 18-May Robert C Lehman     Mets on Sunday (285 chars)
  A    5) 18-May Robert T. Cartolano Mac stuff (5258 chars)
R A    6) 18-May Sue Zayac           SPSSX TNote Draft (251 chars)

[ H=headers R=read REV=review S=send Q=quit BYE ?=Hints HELP ]
MM>
```

* Reading Old Messages

You can reread messages in your mail file with the *read* or *review* command and the message number.

```
MM>headers 6
R A    6) 18-May Sue Zayac           SPSSX TNote Draft (251 chars)

[ H=headers R=read REV=review S=send Q=quit BYE ?=Hints HELP ]
MM>read 6
 Message 6 (251 characters):
Received: by cunixf.columbia.edu (5.54/5.10) id AA12852;
        Fri, 18 May 90 15:55:07 EDT
Date: Fri, 18 May 1990 15:55:05 EDT
From: Sue Zayac <sue@cunixf.columbia.edu>
To: walter
Subject: SPSSX TNote Draft
Message-Id: <CMM.0.90.580161305.sue@cunixf.columbia.edu>

I think this is now ready for critical reading.  Do you
want to find readers or shall I?   /sue
[Press any key to continue]

[ D=delete H=header R=reply TY=type PRI=print Q=quit ?=Hints HELP ]
Read>
```

You can use *read from <name>* and *read subject <string>* and other variants, too. Type ***help message-sequence*** for the variants.

* Forwarding Messages

Messages can be forwarded to other computer IDs with the *forward* command. Before the message is sent on, you are given an opportunity to preface it with a comment. If you don't wish to insert a comment, just press the "Escape" key (the way you usually finish entering text). MM will send the message with a header indicating it has been forwarded from you and not sent directly from the original sender. Replies will come to you.

In this example, Walter sends Sue's message on to two other IDs. He prefaces her message with a note of his own.

```
Read>forward (message to) lynn,lisa
 Message (End with CTRL-D or ESC
   Use CTRL-B to insert a file, CTRL-E to enter editor, CTRL-F to run text
   through a filter, CTRL-K to redisplay message, CTRL-L to clear screen and
   redisplay, CTRL-N to abort, CTRL-P to run a program and insert output.):

Can you read Sue's new TNote before Friday? - Walter

[ D=display ED=edit H=header S=send TY=type Q=quit ?=Hints HELP ]
Send>s
lynn... Queued
lisa... Queued

[ D=delete H=header R=reply TY=type PRI=print Q=quit ?=Hints HELP ]
Read>

[ H=headers R=read REV=review S=send Q=quit BYE ?=Hints HELP ]
MM>exit
Expunging deleted messages.
[1] + Stopped  (signal)          mm
$
```

Walter used the TAB key to make the computer finish out the *forward* command, so "ward (message to)" was printed by the computer. Also, note that when Walter exited MM this time, the message he deleted earlier was expunged.

Lynn and Lisa will receive a message containing Walter's line "Can you read..." followed by Sue's message to Walter.

The command *remail* acts similarly, but does not let you insert any comment, and replies will go to the original sender. *Remail* is for messages that weren't really for you.

* Copying To and From Regular (Non-Mail) Files

COPYING FROM A FILE INTO A MESSAGE

You can insert a regular file into a message. Start as you would in *send*ing a message. When you wish to insert the text from the file, type CTRL-b. You will then be prompted for the name of the file. In this example, Sue sends her document, spssx-tnote.mss, to Lisa and Lynn.

```
MM>send
 To: lisa,lynn
 cc: maurice
 Subject: SPSSX TNote Draft
 Message (End with CTRL-D or ESC
  Use CTRL-B to insert a file, CTRL-E to enter editor, CTRL-F to run text
  through a filter, CTRL-K to redisplay message, CTRL-L to clear screen and
  redisplay, CTRL-N to abort, CTRL-P to run a program and insert output.):

Here it is:
^b
Insert file: spssx-tnote.mss
[OK]

If you have any questions, send me mail.  /sue

[ D=display ED=edit H=header S=send TY=type Q=quit ?=Hints HELP ]
Send>s
lisa... Queued
lynn... Queued

[ H=headers R=read REV=review S=send Q=quit BYE ?=Hints HELP ]
MM>
```

CTRL-b is not visible on screen, just the resulting prompt *Insert file:* .

It is not necessary to type anything to MM before or after inserting an external file; your entire message can consist of the inserted file.

COPYING FROM A MESSAGE TO A FILE

The *list* command copies a message to a regular file. For example, here Lynn has just read Sue's message containing the document, and decides to copy it out to a file, so she can work on it outside MM.

```
Read>list suesdocument

[ D=delete H=header R=reply TY=type PRI=print Q=quit ?=Hints HELP ]
Read>
```

From the *MM>* prompt, specify a message sequence after ***list suesdocument***.

A file created by *list* starts with a list of message headers (like the result of the *headers* command) for its contents, followed by a page break, and then copies of each message. There is no page break between the messages, unless you request breaks by typing *list /separate-pages ....* *List* does not add to an existing file; the new file created by *list* will replace an existing one of the same name.

* File Transfer Using Kermit

Kermit can transfer files between your PC and the host computer that runs MM. File transfer is actually the original purpose of Kermit.

There are numerous reasons to use file transfer in connection with MM. You may want to compose long MM messages with your usual PC word processor, to avoid long connect time and to avoid learning emacs. You may want to send a PC file that wasn't originally intended for electronic mail. You and a colleague may want to work together on an article, using electronic mail to send drafts from one PC to another, possibly over long distance.

FILE-TO-FILE TRANSFER

The basic Kermit operation is to transfer files between the PC and the host. For information, see the handouts *Using MS-DOS Kermit* and *Using MacKermit*, and the booklet *Kermit User's Guide*, at the Business Office, 102 Philosophy Hall (854-3555).

You can use file-to-file transfer if you wish. Downloading, use the MM command *list* at the *MM>* or *Read>* prompt to copy a message into a file on the host, and then, at the shell, use Kermit to transfer that file to your PC. Uploading, use Kermit to transfer a file from your PC to the host, and then, when sending a message, use CTRL-b to copy that file into the message.

The basic procedure has the drawback that a file is created on the host solely for purposes of transfer. You don't really need it there, and you have to remember to remove it at some point. The following examples show how to transfer directly from and to MM.

TRANSFER FROM MM TO PC

In this example, Lynn has just read the document sent by Walter, and decides to copy it to a file to be called *spssx* on her IBM XT.

```
Read>list | "kermit -s - -a spssx"
Escape back to your local system and give a RECEIVE command...
    ^]c

    Kermit-MS>receive
    [--Kermit screen showing file transfer--]
    Kermit-MS>c

Read>
```

The Kermit commands, shown above as indented, are cleared from the screen once the file transfer is complete.

The key to the transfer is the command *list | "kermit -s - -a spssx"*, which starts up the host Kermit and tells it to send. Notice the quotation marks.

Lynn also could have done this later on from the *MM>* prompt. If it were message 12, for example, the command at *MM>* would be *list | "kermit -s - -a spssx" 12*.

TRANSFER FROM PC TO MM

Kermit transfers files in a format called *ASCII* or *text*. Many MS-DOS word processors keep files in their own special formats, so the files must be translated into ASCII before sending and from ASCII after receiving. Each word processor has a procedure to do so fairly simply. (Macintosh is similar. A file you create should be saved as *text*. When you try to open a *text* file for editing, it is converted automatically.)

In this example, Lynn has written up comments about Sue's document using her favorite word processor on her XT. She then used the word processor's commands to make an ASCII text version of the document, named *comments*. Now she is sending it to Sue.

```
MM>send
 To: sue
 cc:
 Subject: spssx tnote
 Message (End with CTRL-D or ESC
  Use CTRL/B to insert a file, CTRL/E to enter editor, CTRL/F to run text
  through a filter, CTRL/K to redisplay message, CTRL/L to clear screen and
  redisplay, CTRL/N to abort, CTRL/P to run a program and insert output.):

Sue, here are some comments on the spssx tnote:

^p
Command: kermit -k
    ^]c

    Kermit-MS>send comments
    [--Kermit screen showing file transfer--]
    Kermit-MS>c

[Done]

Lisa is sending her comments separately.  Lynn

[ D=display S=send TE=text ED=edit TY=type Q=quit ?=Hints HELP ]
Send>
```

CTRL-p does not appear on screen, but only the response to it, the prompt *Command:* . After the command *kermit -k*, the cursor drops to the next line, but there is no prompt at all. After the file transfer, the Kermit commands, shown above as indented, are cleared from the screen, so the comment *[Done]* appears on the line right under *Command:* .

As you can see, MM states that CTRL-p is used "to run a program and insert output". The command *kermit -k* runs the host Kermit and tells it to stand by to receive a file. Kermit's output is the file that it receives.

In the example, Lynn wrote a line of text before and after the file. This is not required; the file could be the entire message.

MAC KERMIT

Mac Kermit works substantially the same as MS-DOS Kermit, shown in the examples here. Instead of typing ***CTRL-] c*** and then ***send*** or ***receive***, pull down the file menu and choose *send* or *receive*.

* Using Emacs from within MM

MM's editor, called text mode, is fairly limited. It does line wrap, but the only way to go back and change anything is to backspace to it, erasing the last part you typed, and then retype forward.

GNU Emacs, a screen-oriented text editor, is far more powerful. If you know *emacs*, you may want to use it from within MM. [Note: If you don't know *emacs*, try the online tutorial. Type ***edit*** to the shell prompt *$*, then type ***CTRL-h t***.] With *emacs* you can do such things as move easily back and forth, insert, move and delete larger pieces of text, clean up messy lines by re-justifying paragraphs, search for and replace specific strings of characters, and use the *ispell* spelling checker and correction program.

It is difficult to demonstrate the use of a screen-oriented editor on a piece of paper, so this example only shows how to begin and end *emacs* and then send the edited file. In this example, Lynn is sending comments on the SPSSX TNote back to Sue. She starts her message in text mode. After a line or two, she decides that she needs more editing power. She then hits CTRL-e to move to *emacs*.

```
$ mm
/usr/local/mm

[ H=headers R=read REV=review S=send Q=quit BYE ?=Hints HELP ]
MM>send
 To: sue
 cc: lisa
 Subject: Comments on the SPSSX TNote Draft
 Message (End with CTRL-D or ESC
  Use CTRL-B to insert a file, CTRL-E to enter editor, CTRL-F to run text
  through a filter, CTRL-K to redisplay message, CTRL-L to clear screen and
  redisplay, CTRL-N to abort, CTRL-P to run a program and insert output.):

I have a couple of comments:

1.  COuld you make the 2nd exxample shorter.

2.  In the first exampel, I think your using one of your
own commands, not a system one.  Tiht may might will confuse
people.

2.  Thirdly.  Oh drat, I better edit this....
^E
```

The display in *emacs* is actually two screens: In the smaller, top screen is the address information. In the larger, bottom screen is the message waiting to be edited. At the very bottom you will see the message *Don't forget to save your buffers if you want your changes to take effect.*

```
From: Lynn Jacobsen <lynn@cunixf.cc.columbia.edu>
To: sue
Cc: lisa
Subject: Comments on the SPSSX TNote Draft
-----Emacs: *MM Headers*        (Text Fill)--------All--------

   I have a couple of comments:

   1.  COuld you make the 2nd exxample shorter.

   2.  In the first exampel, I think your using one of
   your own commands, not a system one.  Tiht may might
   will confuse people.

   2.  Thirdly.  Oh drat, I better edit this....



-----Emacs: *MM Outgoing*       (Text)----Bot-----------------
Don't forget to save your buffers if you want your changes to take effect
```

Now make whatever changes need to be made. When you are finished, start returning to MM by typing CTRL-x CTRL-z:

```
From: Lynn Jacobsen <lynn@cunixf.cc.columbia.edu>
To: sue
Cc: lisa
Subject: Comments on the SPSSX TNote Draft
-----Emacs: *MM Headers*        (Text Fill)--------All--------

   I have a couple of comments:

   1.  In the first example, I think you're using one of
       your own commands, i.e., "pp".  This will confuse
       people.  It would be better to use "p unn".

   2.  Could you make the 2nd example shorter?  You
       don't really need the second "list".

   3.  There's a typo in example 3.  It's "curmudgeon"
       not "crudmugeon".



-----Emacs: *MM Outgoing*       (Text)----Bot-----------------
C-x C-z
```

Immediately after you type CTRL-x CTRL-z but *before* you are returned to MM, you will see a message like this one at the bottom of the *emacs* screen:

```
Save file /f/us/us/lynn/.mm-outgoing.21393? (y or n)
```

Type a ***y*** if you want to save this message. Typing an ***n*** will discard the changes made since starting *emacs*. Actually, *emacs* will ask first if you are sure you want to discard the message.

Now you are back at the *Send>* prompt. If you like, use the *display* command to check what your editing looks like.

If, at the *Send>* prompt, you still had second thoughts about the message, you could type the word ***edit*** to return again to *emacs*.

The message is sent only when you give a ***send*** command.

* Additional Mail Files

Your main mail file is named *mbox*. All the messages you have are stored in the one file. When MM starts up, it copies any new mail you have into mbox, and then shows you the headers of the new mail.

If you want to keep a large number of old messages, it is a good idea to put them into additional mail files, rather than leave them in mbox. MM can start up faster. You might have just one other file for old mail, or you could classify the mail by topic or sender.

Either of the commands *move* and *copy* place mail into another mail file. The difference is that *move* marks the messages for deletion, while *copy* does not.

The command *get* is used to change between mail files.

Here, Sue has decide to start a mail file called waltermail to hold messages from Walter. To be sure, she then *get*s that file and uses the *headers* command to check it.

```
MM>move waltermail from walter
File does not exist: /f/us/us/sue/waltermail
Do you want to create it? yes
 4,15:16,22,26,31:32,45,51

[ H=headers R=read REV=review S=send Q=quit BYE ?=Hints HELP ]
MM>get waltermail

[ H=headers R=read REV=review S=send Q=quit BYE ?=Hints HELP ]
MM>headers all
```

Since this is a new mail file, MM first verifies whether Sue wants to create it. The response to the *move* command then indicates that the messages specified by "from walter" are 4, 15, 16, 22, 26, 31, 32, 45 and 51. In the new file waltermail, they will be numbered 1, 2, 3, etc., as Sue will see in response to the *headers all* command issued at the end of the example. Since Sue used the *move* command, the messages are now marked for deletion in mbox.

From now on, Sue can type ***move waltermail*** at the *Read>* prompt after reading a message, and that message will be moved to the waltermail file.

Once she has typed the command *get waltermail*, Sue can type the usual MM commands and they will apply to the waltermail file. One thing she will need to do is delete unwanted messages in it, and type the *expunge* command to eliminate them, so that the file does not become too large.

To return to mbox, Sue can then type ***get mbox***, but the command *get*, with no filename, always refers to mbox, so Sue actually types:

```
MM>get

[ H=headers R=read REV=review S=send Q=quit BYE ?=Hints HELP ]
MM>
```

* Nicknames and Mailing Lists

NICKNAMES

Probably you will send mail regularly to certain people. If they have long or strange addresses, you may want to define an *alias* for each of them to save typing or to avoid looking up the address. To do so, use the *define* command:

```
MM>define linda muurb@cuvmb
MM>define rich rich@machine1.com.bfu.edu
```

Then, when you send mail, type the alias after *To:*, and MM will send the mail to the real address you defined.

After you have defined an alias, you must deliberately save it if you want to use it again. To save an alias definition, use the *save-init* command. It takes no arguments:

```
MM>save-init
```

This will update a file named *.mminit* in your directory (or create it). Whenever you use MM in the future, any aliases in this file will be available.

To find out what aliases you have already defined, use the command *who ?*. This will give a list of aliases. To find out who a particular alias is, type ***who*** and the name of the alias. For example:

```
MM>who ? mail alias, one of the following:
linda    rich
 or recipient name, text string
MM>who linda
muurb@cuvmb
```

MAILING LISTS

You can also use an alias as a mailing list. Just define the alias as a group of user IDs:

```
MM>define stats sue,maurice,jte,wmbcu@cuvmb
```

Send mail, typing *stats* after the *To:* prompt, and MM will send it to the whole list.

Although the purpose is different, a mailing list alias is the same thing to MM as a nickname alias. The *who* command now will show this:

```
MM>who ? mail alias, one of the following:
linda    rich    stats
 or recipient name, text string
MM>who stats
sue, maurice, jte, wmbcu@cuvmb
```

A better way to define a mailing list is to create a file that has all the addresses in the list. (Use *emacs* for example.) Then use the *define* command to tell MM about the file. In the example below, we start by creating a file called *statspeople* containing all the ID's we want in our group.

The shell command *more* shows the contents of a file.

```
$ more statspeople
sue,maurice,jte,wmbcu@cuvmb.bitnet
```

Now we define *stats* so that it will be an alias for whatever is in the file *statspeople*.

```
MM>define stats @@statspeople
MM>who stats
@@/us/us/lynn/statspeople
MM>send
 To: stats
 cc:
 Subject: Time sheets
 Message (End with CTRL-D or ESC
  Use CTRL-B to insert a file, CTRL-E to enter editor, CTRL-F to run text
  through a filter, CTRL-K to redisplay message, CTRL-L to clear screen and
  redisplay, CTRL-N to abort, CTRL-P to run a program and insert output.):

   There was an error in one of the timesheets last week.
   I will correct it as soon as I can.

   lynn
S>display
From:  lynn jacobsen <lynn@cunixf.cc.columbia.edu>
To:  sue, maurice, jte, wmbcu@cuvmb.bitnet
Subject: Time sheets

There was an error in one of the timesheets last week.
I will correct it as soon as I can.

lynn
```

The symbols @@ tell MM to use the file *statspeople* whenever you mail to *stats*. You can change the file as the people involved change, and each time you send a message to *stats*, MM will look at the file *statspeople* to see who is in it now. Note that *who stats* shows the name of the file, not who is in it.

If, instead, you type *define stats @statspeople*, with only one @, MM will copy the list of names out of *statspeople* at the time you define it. MM will not look in the file again: even if you change it, MM will not change the definition of *stats*. In this case *who stats* will show the list of names.

The file can have the same name *stats* as the alias; we used different names here to clarify the discussion.

3. Message Sequences

* Message Sequences

When you are working in **read mode** or **send mode**, any command you enter will apply either to the message you are reading or to the outgoing message. At **top level**, however, commands can apply to any or all of the messages in your mailbox. You may, for example, want to read all messages from a particular person about a certain subject. Or perhaps you'd like to delete any message about a meeting or class once the event has passed. Message sequences make this possible.

A message sequence is a word or phrase that describes some group of messages that have a particular trait in common.

The simplest form is a number or range of numbers. For example, messages 5 through 10 can be specified as follows: 5:10 5-10 5,6,7,8,9,10 5+6

Asterisk (*) means the last message, the highest-numbered message.

There are many other message sequences, listed below. Some require a value as suggested in <> brackets. Type *help message-sequence* to list them online.

* Description of Message Sequences

- AFTER <date>: all messages sent or received after a certain date or day, as *after April 1, 1989* or *after Tuesday*. A time can be specified after the date or day, as *after Tuesday 1:00pm*.
- ALL: every message in your file, including any marked for deletion, from oldest to newest. See also INVERSE.
- ANSWERED: messages you have *reply*'d to or *answer*ed.
- BEFORE <date>: all messages sent or received before a certain date or day. See AFTER.
- CURRENT: the message MM is pointing at, for example, the last one *read*. Also abbreviated as period (.).
- DELETED: messages that have been marked for deletion with the *delete* command, or moved using the *move* command. On a subsequent *expunge* or *exit* command, the deleted messages will be physically removed from your mail file and gone forever.
- FLAGGED: messages marked by the *flag* command.
- FROM <user>: messages from a person. It searches both the user ID and the name in the *from* field, so note, for example, that *from sue* matches the user ID *sue* but also anyone else named Sue. Be careful when deleting *from* a name.
- INVERSE: all messages, like ALL, but in inverse order (most recent message first).
- KEYWORD <keyword-name>: messages marked by the *keyword* command. You can define keywords to mark groups of messages by a keyword name. Whenever you want to refer to the messages marked with a certain keyword, you type a two word sequence: *keyword <keyword-name>*.

- LAST: the last message. Also abbreviated as asterisk (*).
- LAST <number>: the last *n* messages.
- LONGER <number>: all messages containing more than, or exactly, a certain number of characters.
- NEW: messages that are both *recent* and *unseen*. Generally this is a message that has arrived while you were away from the computer, or perhaps one that has just arrived while you were working.
- ON <date>: messages sent or received on a certain date or day. See AFTER.
- PREVIOUS SEQUENCE: the last message sequence used in an MM command. You might preview a sequence by typing something like *headers <message sequence>*, and then, if it is what you want, type *read previous sequence* rather than retyping the same message sequence again.
- RECENT: messages that are new as of this session with MM. If you have 5 new messages when you log in, then they would all be marked *recent* until you've ended your session with MM.
- SEEN: messages you have already looked at. Most likely all or nearly all the messages in your mailbox are considered *seen* since you usually read any *unseen* ones when you start MM. A message that is marked *seen* will not have a U or N next to its header.
- SHORTER <number>: all messages containing fewer than a certain number of characters.
- SINCE <date>: messages sent or received on or after a certain date or day. See AFTER. Unlike *after*, *since* includes the date specified.
- SUBJECT <word>: all messages that have a certain word (or part of a word) in their subject field. For example, to see messages that have "computer" in the subject, use *subject computer*.
- TEXT <word>: all messages that have a certain word (or part of a word) in their text. For example, the word "computer" might not be in the subject field of some messages, but you can reference all messages that mention "computer" in their text by *text computer*. You might try part of the word instead, like *text comput*, to include "computing", "compute", etc as well.
- TO <user>: all messages that were sent to a user, including carbon copies, but not blind carbon copies (the "cc" but not "bcc" header fields). Most of the messages in your mailbox are *to* yourself, but you may have some *to* other people that are cc'd to yourself, and some that were *to* other people besides yourself.
- UNANSWERED: messages you did not *reply* to or *answer*.
- UNDELETED: messages that haven't been deleted via the *delete* or *move* commands.
- UNFLAGGED: messages not *flag*ged.
- UNKEYWORD <keyword-name>: messages not included in the keyword.
- UNSEEN: messages that were never *type*d or *read*.

* Message Sequence examples

Here are some simple examples of commands using message sequences. You may also use several in one command to further specify the desired messages.

This would put you into **read mode** and would let you read all the *flag*ged messages:

```
MM>read flagged <cr>
```

This would read all the messages from hmh:

```
MM>read from hmh <cr>
```

This would read messages 3 through 8 individually:

```
MM>read 3:8 <cr>
```

This would display the header lines of all the messages that are unanswered from hmh (note that two message sequences are combined, *from* and *unanswered*):

```
MM>hea from hmh unanswered <cr>
```

This command marks for deletion all messages from 5 through 15 that contain the string "deadline" in the text of the message:

```
MM>delete 5:15 text deadline <cr>
```

This would display the header lines of all messages you have received since (and including) Monday:

```
MM>headers since Monday <cr>
```

4. Top Level

* What is Top Level?

When you start MM you are at the **top level**. This is identified by the *MM>* prompt being displayed. At this level, all the power of MM is available to you via a large number of commands that enable you to manage your messages, read them, send new ones, reply to old ones and so forth.

This chapter describes the functions and use of each command. At the *MM>* prompt, you can type *help <command-name>* for more information on each command, with examples.

Of all the **top level** commands, by far the ones you will use most frequently are *read* and *send*. They each start up their own mode that is identified by a different prompt from the **top level** prompt, *MM>*. Some of the same commands can be used. See the following chapters on **read mode** and **send mode**.

* Commands in Top Level

Some of the descriptions that follow mention "arguments". When what you type is of the form *command something*, for example *reply 3*, *reply* is the command and *3* is the argument. The argument that *reply* takes is a message sequence. As you recall from the earlier description of a message sequence, the argument might also be *4:10* or *2,5,7* or *from hmh* instead of *3*. When you are using MM, you can type *?* for suggested arguments, like ***reply ?***.

The type of argument for each command is suggested here in brackets <>. <Mess-seq> is any message sequence; if none is given the command refers to the "current" message. The argument may be optional in some cases, as noted.

- ANSWER: See *reply*.
- BACKTRACK: Not yet functioning. It will be used in conjunction with the *follow* command to track a "conversation". A conversation in this case is considered to be all messages with the same subject. *Backtrack* moves you to the previous message in the conversation.
- BLANK: Blanks your screen if you are on a video display terminal. If not, then it does nothing.
- BROWSE <mess-seq>: Displays headers of the messages specified. It then allows you several options including reading each message, replying to it, flagging it, and deleting it. Type ***?*** during *browse* to see a list of the options.
- BUG: Can be used to report problems or suggestions to the maintainers of MM. *Bug* puts you into **send mode** with a predefined list of addresses to send the message about the bug. You send it the normal way you would send any other message. The response may take a short while, i.e. a day or two. Note, for help and advice on using MM, send mail to "consultant" instead of using *bug*.
- BYE: Exits you from MM and kills the process. It also asks you whether you would like to expunge deleted messages. See EXIT and QUIT.
- CD <directory>: The *cd* command is similar to the *cd* command in the shell. It changes the current working directory to one specified as an argument. The default directory is your home (login) directory. The current working directory is the starting point for path names not beginning with "/".

- CHECK: Checks for new messages that may have arrived while you are using MM. MM does an automatic check every 5 minutes.

- CONTINUE: If you quit at the *Send>* prompt, *continue* returns you to the *Send>* prompt.

- COPY <filename mess-seq>: Copies messages to an additional mail file. It takes two arguments: the first is the filename of the additional mail file, and the second is a message sequence. The messages are left unchanged in the original mail file. If the file to copy to does not exist yet, it is created. See MOVE.

- COUNT <mess-seq>: Accepts a message sequence (default *all*) and displays the message numbers and total number of messages in the sequence.

- DAYTIME: Tells you the current date and time.

- DEFINE <alias user(s)>: The *define* command is used to create nicknames or mailing lists, known in MM as aliases. The two arguments are first, the alias you will use, and second, the real address (which may be a list of users, separated by commas). See page 22 for an example. You can also refer MM to a file listing the users, as explained at the example. To keep the alias permanently, use the *save-init* command.

- DELETE <mess-seq>: Marks a message or set of messages for deletion. The messages are not erased until the *exit* or *expunge* command is given.

- ECHO <text>: Prints the same text back at the terminal. Useful in files of mm commands that you would *take*. See TAKE.

- EDIT <mess-seq>: Edits a message or set of messages in your mail file, using the editor specified in your *.mminit* file. The default editor is *emacs*.

- EXAMINE <filename>: Changes your current mail file to an additional file in read-only mode. It is like the *get* command except that the file is read-only, and the file reference date is not updated.

- EXIT: Ends your current MM session and suspends the process. It will also erase any messages you have marked for deletion in the current mail file (the one you are in when you *exit*). See BYE and QUIT.

- EXPUNGE: Permanently eliminates messages marked for deletion. It actually writes out a new copy of the file without the deleted messages, so in effect it makes the file shorter. The command *exit* does an *expunge* for you.

- FINGER: The *finger* command is the same one available to you at the shell. It shows you information about other users on the system. See the examples on page 45. For further information, type *man finger* at the shell prompt, or *!man finger* at any of the MM prompts.

- FLAG <mess-seq>: Makes the messages specified "stand out" in your mail file by marking them as flagged. You can refer to them with the message sequence *flagged*. *Flagged* messages are also displayed when you start MM, along with the unseen messages.

- FOLLOW: Not yet functioning. It will be used in conjunction with the *backtrack* command to track a "conversation". A conversation in this case is considered to be all messages with the same subject. *Follow* moves you to the next message in the conversation.

- FORWARD <mess-seq>: Forwards a message you have received to some other address. It is similar to REMAIL, but it allows you to insert a message on top of the one being forwarded. You are the "sender" of the forwarded message, so replies come to you.

- GET <filename>: Changes your current

mail file to the file named. All MM commands now apply to that file. With no argument, refers to your main mail file. See EXAMINE.

- HEADERS <mess-seq>: Lists "headline" summaries of messages in your current mail file. If you follow this command with a message sequence, it will output all the headers of that sequence in order of lowest message number to highest. See page 10 for a detailed explanation of each of the fields that appear when you use this command.

- HELP: Displays help on various topics including all the commands at the level at which *help* is being invoked. To find out the things you can get help on type *help ?*. You can also type *help <command>* for help on that command.

- JUMP <message>: Resets a certain message to be the *current* message. For instance, *jump 10* will make message number 10 be the current message.

- KEYWORD <word-list mess-seq>: Allows you to group together related messages by assigning them keywords. The first argument is one or more keywords separated by commas, and the second argument specifies what messages to assign them to. Later, when you use the message sequence *keyword <word>*, it refers to all messages to which you gave that keyword.

- LIST </switch filename mess-seq>: This use of *list* formats messages nicely and copies them to a file. The difference from *copy* is that the file cannot be read by MM. There are three arguments. The first is optional, either */headers-only*, which lists only the header (meaning what you would see from a *header* command), or */separate-pages*, which puts a page break between messages. The second argument is the name of the file. The last argument is a message sequence.

- LIST </switch | "shell-command" mess-seq>: This use of *list* formats messages nicely and sends them as input to a shell command. There are three arguments. The first is optional, either */headers-only* or */separate-pages*, as described above. The second argument is a pipe symbol (|) followed by any shell command, in double quotes, that will take the input. The last argument is a message sequence. This use of *list* can be used to print messages by putting the appropriate shell print command in quotes (like "lpr -P<printer>"). It can also be used to send the message into Kermit so you can copy it to a file on your PC; see page 16.

- LITERAL <command>: *Literal* is a prefix to the *list*, *print* and *type* commands that causes them to ignore the *dont-print-headers*, *only-print-headers*, *dont-type-headers* and *only-type-headers* variables.

- MARK <mess-seq>: Marks a message or set of messages as **seen**.

- MOVE <filename mess-seq>: Moves messages into an additional mail file. It works like *copy*, but also marks the messages for deletion from your current mail file. See COPY.

- NEXT: Goes to the next message in the file and types it if it is not deleted. The next message is considered to be the one directly after the current one.

- PREVIOUS: Goes to the previous message in the file and types it if it is not deleted. The previous message is considered to be the one directly before the current one.

- PRINT </switch mess-seq>: Prints the specified messages using the print program defined by the *print-filter* variable. The first argument is optional, and the only possibility is */separate-pages*, which puts a page break between messages, so each starts on a new page. The default *print-filter* is */usr/local/bin/print*, which will ask what printer you want to use. If you connect via Kermit from a PC with its own printer directly attached, try the command *set print-filter pcprint*, and *print* will then direct printing to your PC's printer. As an example of the full command, *print /separate-pages 6:10* prints messages 6 to 10 with each on its own page.

- PROFILE: Helps you set up an environment for using MM corresponding to your preferences in message handling. It asks you a series of questions and then makes MM remember your responses (via the *.mminit* file). It does not go through all of the *.mminit* options possible. For further details see the chapter on **Customization**.
- PUSH: Gives you a new (inferior) shell. At that point you can then do anything you could ordinarily do in the shell, and you get back to MM by exiting the shell. A different option is to enter shell commands at any of the MM prompts simply by preceding them with a "!", e.g. *!man finger*. Still another option is to quit from MM by *quit* or *exit* and then return to MM afterwards.
- PWD: The *pwd* command is similar to the *pwd* command in the shell. It displays your current working directory. See the *cd* command to change your working directory.
- QUIT: Quits out of MM, without expunging, and suspends the process. See BYE and EXIT.
- READ <mess-seq>: Starts reading the messages specified in **read mode**. For more information see the chapter on **read mode**. Without an argument, *read* will read any messages you have not yet seen.
- REMAIL <mess-seq>: Similar to *forward*, but the message is sent as is, with nothing added except modification to the header fields to indicate who did the remailing. Each message in a sequence is sent as a separate piece of mail. The original sender will get any replies.
- REPLY <mess-seq>: Most commonly used in **read** mode to reply to the current message, but can also be commanded from top-level. With no argument, refers to the current message. When invoked from **top-level**, it will ask who to reply to after you type the command *send*: type *sender*, *all* (meaning everyone who received the message), or *none* (meaning don't send a reply to this message); also type *including* to include the original message in the reply, for example *sender including*.
- RESTORE-DRAFT <filename>: Continue sending a message you saved in mid-composition using the *save-draft* command (page 37).
- REVIEW <mess-seq>: Like *read*, but when no message sequence is specified, review prompts you for one.
- ROUTE <user>: Forwards all mail sent to you to another address. The address can be either a local user ID or a remote mail address in the form *user@host*. To stop the forwarding, type *route* with no argument. The command creates a file called *.forward* in your directory.
- SAVE-INIT: Used after one or more *define* or *set* commands to save the definition or setting permanently in the file called *.mminit* . See DEFINE and the chapter on **Customization**.
- SEND: Start writing a message to be sent. See the chapter on **send mode**. An alternative format is *send <user>*, specifying the user you are sending to.
- SET <variable-name value>: Customize MM to act differently, by resetting one of its variables. See the chapter on **Customization**.
- SHOW <variable-name or mail-alias>: With no argument, displays the current MM environment variable settings and

mail aliases, as established by the *set* and *define* commands. You can also specify a particular variable or alias to see what value it has currently. See DEFINE and the chapter on **Customization**.

- SORT: Sorts the mail file chronologically by the dates the messages were generated.

- SPELL <mess-seq>: Invokes the program indicated by the *speller* variable on the message specified. (The default speller is *ispell*.) You should *spell* one message at a time. Type *man ispell* in the shell (or *!man ispell* at any of the MM prompts) for more information. Once you are in *ispell*, type *?* for help or *X* to exit without changing the file.

- STATUS: Tells you relevant information and statistics about your current message file, i.e. how many messages are deleted, unseen, how large the file is, etc. Typing *status verbose* gives you in addition the process ID number and user name.

- SUSPEND: Suspends execution of MM. It may be continued later with the shell *fg* (foreground) command. Modified mail files will be saved before control is returned to the shell.

- TAKE <filename filename filename>: Directs MM to an external file of MM commands and executes the commands. It takes three arguments: the first is the file containing the commands; the other two are optional and are for output and error messages respectively. If no output file is specified, the commands themselves will be invisible and only the results of the commands will appear. If you define the output file as */dev/tty*, the commands themselves will be displayed at your terminal as well as the output from the commands. MM closes the file and restores input from the terminal when any of the following happen: end of file, command error, or a *take* command with no argument (this suppresses the "[End of ...]" message). Note, each time MM starts up it automatically *take*s the files *.mminit* and *.mmrc* in your login directory.

- TYPE <mess-seq>: Displays the messages specified without going into **read** mode.

- UNANSWER <mess-seq>: Removes the answered status from messages.

- UNDELETE <mess-seq>: "Undeletes" messages; that is, it removes the marker that calls for deletion. (Expunged messages no longer appear in *headers* and cannot be "unexpunged".)

- UNKEYWORD <word-list mess-seq>: Removes specified keyword(s) from the specified message(s).

- UNFLAG <mess-seq>: Removes the flag, as set by *flag*.

- UNMARK <mess-seq>: Unmarks messages; that is, makes them appear unseen.

- VERSION: Shows MM's current version number, copyright notice, and bug report address.

- WHO <mail-alias or user>: Shows how a given mail address will be translated by the mail system into a list of one or more actual recipients. With an alias, shows the users (or the file containing a list of users). With a user ID, usually shows the user ID, but will show any forwarding or other re-addressing that may be in effect.

- WRITE <filename>: Writes out a new copy of the mail file. It does not remove deleted messages. With the optional filename argument, it will make a copy of your mail file with the new name.

*** CCMD**

At any MM prompt, you are using a command parser called CCMD, which you can use to help you enter commands.

Partially typed commands can be completed by typing *[TAB]*. For example, to issue the command *check*, you could type just *ch* and then *[TAB]*. The reason this works is that only one command begins with *ch*. Try typing *c[TAB]*: it will be completed too, but not perhaps as you wish, since there is more than one command starting with *c*; abort with *<ctrl/u>*, or backspace over it.

If the partially typed command takes an argument, CCMD will describe what it is. For example, if you type *cop[TAB]*, CCMD fills it out as *copy (into file)*, meaning you should fill in a file name.

To see whether a command has a default argument, type *[TAB]* where you would normally type the argument. For example, *headers [TAB]* is completed as *headers current*. If you want the default, just press *[RETURN]*; otherwise, abort with *<ctrl/u>*, or backspace over it and type what you want instead. If there is no default, your terminal will beep.

Lastly, CCMD performs the *?* help function. At almost any point, you can type ? to see a list of suggestions. Sometimes it just reminds you how to type out a command, and other times it may show you something you never heard of that looks useful. Follow up by typing *help <command>* for more information.

5. Read Mode

* What Is Read Mode?

Read mode reads and processes the messages in your mail file. Its commands are very similar to the ones in **top level**, but refer only to the message being read. So, for instance, if you were to type *MM>**read 3***, you would be put into **read mode**, and commands would then, for the most part, apply only to message 3. If you had typed *MM>**read 3:10*** (meaning to read messages 3 through 10, individually and one right after the other), any commands you would type in **read mode** would apply to whichever message between 3 and 10 is currently being read.

* Commands In Read Mode

Start **read mode** by typing *MM>**read*** followed by a message-sequence. With no message-sequence, *MM>**read*** is taken to mean *MM>**read unseen***, and it will let you read each new message you have not seen. If there are no new messages, it simply returns you to the **top level**. When you start MM, it tells you whether you have new messages. During a session, MM checks for more incoming mail every 5 minutes, or you can force a check with the *check* command.

Most of the top-level commands are also available in **read mode**. **Read mode** is identified by the *Read>* prompt. You can get descriptions of the **read mode** commands by typing *help ?* or *help <command>* at the read prompt. The major difference is that the commands do not take message sequences as arguments, but refer instead to the current message. For example, *delete* refers to the message you just read.

The command *kill* is the only **read mode** command that does not also operate at **top-level**. *Reply* works somewhat differently in **top-level** and **read modes**.

- KILL: Combines *delete* and *next*: marks the current message for deletion and then types the next message.
- REPLY: Most commonly used in **read mode** rather than **top-level**. In **read mode**, the two optional keywords for *reply* should be added immediately after the command. The simple command *reply* is the same as the full command *reply sender not-including*. The alternative to *sender* is *all*, which sends the reply to everyone who received the original message. The second keyword can be *including*, which includes the text of the original message above the reply. Therefore, the following commands are all commonly used: ***reply***, ***reply all***, ***reply including***, ***reply all including***.

6. Send Mode

* What Is Send Mode?

Send mode is invoked by typing the *send* command at **top level** or in **read mode**. Whenever you *send* or *forward* a message, *reply* or *answer* a message, or send a *bug* report, **send mode** is invoked in some form. Suppose you type *send* followed by a <*cr*>:

```
MM>send <cr>
 To:
```

At this point, MM is awaiting a list of addresses to send the message to. Valid addresses are of the form *userID*, *userID@host*, or *userID, userID, userID* (i.e., several user ID's separated by commas). Suppose you want to send a message to hmh, jhs and jcs. You would say

```
MM>send <cr>
 To: hmh, jhs, jcs <cr>
```

The list of addresses is terminated with a carriage return. Notice that spacing and case will be ignored. You could just as well have said *hms,jhs,jcs*. They are separated by space here only for readability. This is known as specifying the *to-list* in sending a message. The next thing it will ask you for is a *cc-list*:

```
MM>send <cr>
 To: hmh <cr>
 cc:
```

The *cc-list* is optional. You can put in one or more valid userIDs in the same form as in the *to-list*. Cc's are used to let someone see a copy of a message that isn't really directed to them. Suppose you have a message for hmh and mmc, and you want vla to be aware that you sent it:

```
MM>send <cr>
 To: hmh, mmc <cr>
 cc: vla <cr>
 Subject:
```

The last prompt is for the subject of the message. While it is optional, you should always use it, by putting in a few words or a phrase that describes the topic of your message. This is what your readers will see as the subject when they see they have new mail. For example, here is what you might say if if you are sending a message to the above people about the recent difficulties you've had getting in touch with them:

```
MM>send <cr>
 To: hmh, mmc <cr>
 cc: vla <cr>
 Subject: Problems calling you <cr>
 Message (End with CTRL/D or ESC
  Use CTRL/B to insert a file, CTRL/E to enter editor,
  CTRL/F to run text through a filter, CTRL/K to redisplay message,
  CTRL/L to clear screen and redisplay, CTRL/N to abort, CTRL/P to run a
  program and insert output.):
```

Now you can actually type in the body of the message itself. Suppose you want to send a message to the above people, saying *I am having problems getting in touch with you on a regular basis. Is there a good time to call you at home or another number I can use?*. To do this you should type:

```
MM>send <cr>
 To: hmh, mmc <cr>
 cc: vla <cr>
 Subject: Problems calling you <cr>
 Message (End with CTRL/D or ESC
  Use CTRL/B to insert a file, CTRL/E to enter editor, CTRL/F to run text
  through a filter, CTRL/K to redisplay message, CTRL/L to clear screen and
  redisplay, CTRL/N to abort, CTRL/P to run a program and insert output.):
I am having problems getting in touch with you on a regular basis.
Is there a good time to call you at home or another number I
can use?<cr>
[ESC]

Send>send
hmh... Queued
mmc... Queued
vla... Queued
MM>
```

The message is typed in and ended with *[ESC]*. It is then sent with the *send* command. MM responds by confirming the delivery to hmh and mmc and vla. Thereafter, it returns you to the **top level** *MM>* prompt.

This message was typed using MM's text mode. See page 18 for information on using Gnu-Emacs as the editor.

The following commands can be used while you are entering text:

- *<ctrl/b>*: insert a file into the place where you are currently typing. You will be prompted with *Insert File:* at which point you should type the name of the file you want inserted and then a carriage return. You can then continue typing more of the message or send it.
- *<ctrl/c>*: clobber MM and return to the shell, heralded by the $ prompt. When you type *<ctrl/c>*, mm will prompt you by asking *Do you really want to exit MM?*. In most cases you should type ***no*** and use the standard methods to exit MM.
- *<ctrl/e>*: invoke the editor specified in the EDITOR variable entry in your *.mminit* file. If you already started typing the message, the text is carried into the editor. The default is gnu-emacs.
- *<ctrl/f>*: runs the message through a filter: uses what you have typed as input to some program and replaces it with the output from the program.
- *<ctrl/k>*: types out the text of the message as it appears thus far.
- *<ctrl/l>*: clears the screen, then types the text of the message as it appears thus far.
- *<ctrl/n>*: aborts the message (defaults to asking before aborting the message).
- *<ctrl/r>*: retypes the line that you are typing, from leftmost character to rightmost.
- *<ctrl/p>*: runs a program from the shell and appends the output to the current message.
- *<ctrl/u>*: erases the line that you are typing, from leftmost character to rightmost.
- *<ctrl/w>*: erases the last word you typed.

- *<delete>* or *<backspace>*: delete the last character.
- *[ESC]* or *<ctrl/d>*: escape to **send mode** heralded by the *Send>* prompt at which point any of the **send mode** commands apply. To send the message, type *send* and then a carriage return.

* Commands In Send Mode

As you see above, simply typing ***send*** at the *MM>* prompt will not immediately put you into **send mode** with the *Send>* prompt. What will do that is finishing the text of your message with an *[ESC]* or *<ctrl/d>*.

However, if you want to change some attribute (like add or subtract an address from the header, change the subject, etc.) you will need to get to the *Send>* prompt by *[ESC]* or *<ctrl/d>*. Then you can use any of the **send mode** commands. At the send prompt you also can specify header fields such as: *Bcc*, *Cc*, *Fcc*, *From*, *In-reply-to*, *Reply-to*, *Subject*, *Text*, *To*, *User-header*.

Several commands are available only in **send mode**. Those commands that are the same as the top-level command generally refer to the message being composed (rather than the current message of your incoming mail file). The exceptions to this are *headers* and *type*, which still refer to the current message.

The following **send mode** commands differ from top-level commands:

- DISPLAY <field>: Displays the message you've typed so far, with its header fields (address, subject). If the message is very long and you just want to display one of the fields, you can type ***display to*** or ***display cc*** or ***display subject*** or ***display text***. Following are the optional fields you can use with the *display* command. Of these, *display header* and *display all* are most useful.
  - *all*: shows the entire message and headers.
  - *bcc*: shows only the blind carbon copy recipients
  - *cc*: shows only the carbon copy recipients
  - *fcc*: shows only the file name that will receive a copy of your outgoing message
  - *from*: shows only the *from* address
  - *header*: shows only the message headers (*To*, *From*, etc.)
  - *reply-to*: shows only the *reply-to* field
  - *subject*: shows only the *subject*
  - *text*: shows only the *text*
  - *to*: shows only the *to* recipients
- ERASE <field>: Erases completely a field of your message, like *to*, *cc*, *subject*, *text*, or *all*. For example, to erase the text of your message, enter ***erase text***. Note, logically enough, if by erasing you end up with no *to* or *cc* field, you will have to specify some address(es) with a ***to*** or ***cc*** command in order to send the message. *Erase* cannot erase the sender field. Here are the fields you can follow the erase command with:
  - *all*: erase the entire message
  - *bcc*: erase the blind carbon-copies address list
  - *cc*: erase the carbon copies address list
  - *fcc*: erase the field containing the file name the message was to be copied to
  - *from*: erase the *from* address
  - *in-reply-to*: erase the *in-reply-to* field.
  - *reply-to*: erase the *reply-to* field
  - *subject*: erase the *subject*
  - *text*: erase the text of the message
  - *to*: erase the *to* address list
  - *user-header*: remove some user defined header
- INSERT <filename>: Inserts the contents of a file as an addendum to

your message. For example, to insert the file *Addendum*, you would type ***insert Addendum***.

- BCC <user>: Sends a blind carbon copy, which is like a carbon-copy except no one who receives the message sees the list of bcc recipients. This is useful if you want to send a note to someone and don't want the person to know that someone else is also receiving a copy of the message. For example, ***bcc brennan***.
- CC <user>: Adds more carbon-copy addresses to the *cc-list*. To add jhs and jcs, you would type ***cc jhs, jcs***.
- FCC <filename>: Defines a file name or set of file names that will receive copies of your outgoing message. This will not appear in the outgoing message. The file will be in current directory unless you specify a path.
- IN-REPLY-TO <text>: Takes a line of text as an argument to make the in-reply-to field of the message you are sending. This field is intended to reflect which message you are answering. It will be generated automatically by MM when you use the *reply* or *answer* commands.
- REPLY-TO <user>: Specifies the *Reply-to* field for messages. This directs replies to a different user ID from the one you are using. Possibly you have more than one ID or address and want the replies to go to one of the other IDs, or possibly you want someone else to collect the replies. The command refers to the message being sent. If you have done *set default-reply-to*, then use *reply-to* followed by null to remove the reply-to field from a particular message.
- SAVE-DRAFT <filename>: Saves the message text in the file you name. The text can be recovered later with *restore-draft* (page 30). This lets you interrupt sending the message but keep the text, so that you can resume at some other time.
- SUBJECT <text>: Replaces the subject of the current message you are sending. To change the current subject to be *Budgets*, you should type ***subject Budgets***. The subject, of course, can consist of more than one word, if you so desire.
- TEXT: If you have gotten to the *Send>* prompt and then want to add more text on to message, you can use this command. You may look at the text already in the file by typing *<ctrl/l>* or *<ctrl/k>* or have it appear automatically every time by changing the setting of the *display-outgoing-message* variable to yes. To add *This is a test* you would simply say

```
Send> text <cr>
This is a test [ESC]
Send>
```

- TO <user>: Adds more addresses to the *to-list*. For example, to add jem, mmc, rdl to the *to-list*, you would simply type

```
Send>  to jem, mmc, rdl <cr>
```

7. Customization

* How to Customize MM

Various options of MM can be changed to customize it. They all start out with either default values or no value, and you do not have to change any of them to use MM. This manual describes MM behavior based on the default values. Once you have worked with MM, you may want to try changing some of them.

Four commands are important in customizing:

- SET <variable-name value>: *Set* changes the particular option to the value you want.
- SAVE-INIT: *Save-init* makes the *set* permanent by saving the new value in the file *.mminit* .
- SHOW <variable-name>: *Show* displays the current value of the variable. With no variable name it shows all of them.
- PROFILE: *Profile* takes you through some of the most likely options and asks which way to set each one. It does what *set* and *save-init* do.

Many of the variables take values meaning yes (*always*, *ok*, *true*, *yes*) or no (*false*, *never*, *no*), or the value *ask*, which means to prompt you for yes or no each time. Some variables instead take text or other values.

For example, to change the setting of the *append-signature* variable to yes, you would enter:

```
MM>set append-signature yes
```

* Description of Variables

The following are the variables and the optional settings available followed by the system defaults. Many of these settings are fairly obscure and you will not find much need to change them. For longer descriptions of each variable, type *help set <variable-name>*.

The most commonly changed variable settings are indicated by checkmarks.

- ALIASES-USE-GROUPS: If yes, aliases defined using the *define* command show only the alias name in the *to* field of outgoing messages. If no, all the user ID's that the alias has included will be listed. The default is no.
- ✔ **APPEND-SIGNATURE**: If yes, the *.signature* file in your home directory will automatically be added to the bottom of your outgoing mail. The default is no.
- AUTO-STARTUP-GET: If yes, MM will get your mail file for you automatically upon entry. The default is yes.
- AUTOWRAP-COLUMN: The numeric argument specifies the column at which to perform automatic word wrapping while collecting a message. If the number is positive, it indicates the absolute column at which to wrap. If it is negative, it indicates the number of column spaces from the right side of the screen. If it is zero, no wrapping is done. The default is -7.

- BROWSE-CLEAR-SCREEN: If yes, the browser will clear the screen at every opportunity. The default is yes.
- BROWSE-PAUSE: If yes, always pause between messages when browsing. The default is yes.
- ✔ **CHECK-INTERVAL**: Defines the time in seconds between each check for new mail. The default is 300 (5 minutes). 0 means never.
- CLEAR-SCREEN: If yes, the screen is cleared at startup and between messages. The default is yes.
- CONTINUOUS-CHECK: If yes, checking for new mail is also done in *send* and *read* modes. The default is no.
- CONTROL-D-AUTOMATIC-SEND: If yes, send message automatically on *<ctrl/d>* from text mode. The default is no.
- CONTROL-E-EDITOR: If yes, *<ctrl/e>* invokes the editor in message collection mode. The default is yes.
- CONTROL-L-CONFIRM: If yes, typing *<ctrl/l>* will enter a *<cr>* as well as clear the screen: if there is a command typed at the prompt when you type CTRL-L, the screen will be cleared and the command will be executed. The default is no.
- CONTROL-N-ABORT: If yes, then *<ctrl/n>* aborts the current command. If ask, then *<ctrl/n>* should ask before aborting. If no, then *<ctrl/n>* never aborts. The default is ask.
- CRT-FILTER: Gives the program to use to display messages one screen of text at a time. The default is /usr/ucb/more -x (the more program).
- DEFAULT-BCC-LIST: Defines a list of recipients that should always be included in the bcc: (blind carbon copy) header field. There is no default.
- ✔ **DEFAULT-CC-LIST**: Gives a list of addresses that will always be in the cc (carbon copy) header field of your outgoing messages. There is no default.
- ✔ **DEFAULT-FCC-LIST**: Defines a file name (or list of file names) in which to put a carbon copy of all outgoing messages. There is no default. You should specify the path, because otherwise the file is always assumed to be in the current directory.
- DEFAULT-FROM: Defines what will go in the from field in outgoing mail. In addition, your user ID will appear in the *Sender:* field for outgoing mail. There is no default.
- DEFAULT-MAIL-TYPE: Indicates the mail format for new or empty files. There are currently four mail formats available: babyl, mbox, mtxt and MH. The default is mbox.
- DEFAULT-READ-COMMAND: Defines a command to automatically execute when you type *<cr>* while reading messages. The default is *next*.
- DEFAULT-REPLY-TO: Defines a *reply-to:* field automatically in outgoing mail. There is no default.
- DEFAULT-SEND-COMMAND: Defines the command to automatically execute when you type *<cr>* at the *send*> prompt. There is no default.
- DIRECTORY-FOLDERS: This is only relevant when using MH format mail files. This variable will allow MM to not add a slash when trying to complete a mail folder name on the command line. This is because the MH format uses a directory for a mail folder. The default is no.
- DISPLAY-FLAGGED-MESSAGES: If yes, display *flagged* messages when the mail file is read into MM. The default is yes.
- ✔ **DISPLAY-OUTGOING-MESSAGE**: If yes, the contents of the message you've already composed will be displayed when you re-enter text mode. The default is no.

- DONT-PRINT-HEADERS: Gives a list of header fields that should not appear when you use the *print* or *list* commands. There is no default.
- ✔ **DONT-TYPE-HEADERS**: Gives a list of header fields that should not be shown when displaying messages. There is no default. See page 44 for an example.
- EDITOR: Indicates the editor you want to use when creating or editing messages. The default is /usr/local/bin/emacs.
- ESCAPE-AUTOMATIC-SEND: If yes, then the message will automatically be sent from send-text mode when *[ESC]* is typed. The default is no.
- EXPUNGE-ON-BYE: If yes, MM will automatically expunge the messages marked for deletion when you exit using the *bye* command. The default is ask.
- FAST-INIT-FILE: If yes, then use fast format initialization file, which means write the file *~/.mmfast* as a summary of your *.mminit* file. The default is yes.
- FINGER-COMMAND: Defines the command to execute when the *finger* command is called. The default is *finger*.
- GNUEMACS-MMAIL: If your editor variable is not *emacs*, then this variable doesn't apply. If it is *emacs* and the value of this variable is yes, it will use the mmail mode in the editor gnuemacs automatically. This gives you a window for the headers or message to which you may be replying and a window for the text of your message. The default is yes.
- HEADER-OPTIONS-FILE: Gives the name of a file that contains header fields you would like to appear in all your outgoing mail. For example, if you want to make a header *Office: 816 Watson*, you may first create a file called *.mm-headers*, with the file consisting of the header line. Then say *set header-options-file .mm-headers*. All your outgoing messages will then bear the *Office...* header. There is no default for this variable.
- INCOMING-MAIL: Gives the file name of the mailbox that MM should check for new mail. The default is /usr/spool/mail/*your user ID*.
- KEYWORDS: Defines a list of keywords you may use in conjunction with the *keyword* command. By defining keywords here, you will be able to use command completion in assigning your keywords when you use the *keyword* command. There is no default.
- LIST-INCLUDE-HEADERS: If yes, the index headers (like *headers* command) for each message will be listed out before displaying messages using the *list* command. The default is yes.
- LIST-ON-SEPARATE-PAGES: If yes, put a form feed between each message listed while printing a group of messages. The default is no.
- MAIL-DIRECTORY: Defines the directory in which to keep your primary mail file and related files. The default is ~, your home directory.
- MAIL-FILE: Defines the file in which to store incoming messages (new mail). The default is mbox.
- MMAIL-PATH: Defines the file name for the gnuemacs mmail library. This is not for general usage. There is no default.
- MODIFY-READ-ONLY: If yes, then when you are using a read-only mail file (as when using the *examine* command) you can modify the file in the buffer (the disk is still read-only). Otherwise, you will get an error message when you try to use any command that marks the messages (i.e. read, delete, etc.) The default is yes.

- MOVEMAIL-PATH: Defines the path for the MM program to move mail from the spool directory to your home directory. The default is /usr/local/lib/mm/movemail .
- NEW-FILE-MODE: The numeric argument is the octal default file mode for files created by *move*, *copy*, etc. The default of *600* gives the highest level of protection, which is read/write by owner only.
- ONLY-PRINT-HEADERS: Gives a list of header fields that would be the only ones shown when you use the *print* or *list* commands. There is no default.
- ONLY-TYPE-HEADERS: Gives a list of header fields that would be the only ones displayed during message typeout. There is no default.
- ✔ **PERSONAL-NAME**: Gives a string for your name as it should appear in outgoing mail in the *from* field. There is no default. So if you don't enter a *personal-name*, your user ID and name from */etc/passwd* will appear in that field.
- ✔ **PRINT-FILTER**: Defines the program to use to print messages. The default is /usr/local/bin/print. If you are using Kermit on a PC and have a printer directly hooked up to it, try setting it to pcprint.
- PROMPT-FOR-BCC: If yes, prompt for blind carbon copy during send command. The default is no.
- PROMPT-FOR-CC: If yes, prompt for carbon copy during send command. The default is yes.
- PROMPT-FOR-FCC: If yes, prompt for the name of a file to automatically put a copy of your outgoing message into during the send command. The default is no.
- PROMPT-RCPT-ALWAYS: If yes, always prompt for the recipients when sending or replying. The default is no.
- ✔ **READ-PROMPT**: Defines the string to be used as the read prompt. The default is *Read*>.
- ✔ **REPLY-ALL**: If yes, replies go to all recipients (not just sender). The default is no.
- ✔ **REPLY-INCLUDE-ME**: If yes, when you are replying to a message and say *reply all*, replies go to your user ID also. The default is no.
- REPLY-INDENT: Gives the indentation string to be used for the included message when including original message into reply. The default is "> ".
- REPLY-INITIAL-DISPLAY: If yes, display the headers associated with the reply to a message after the *reply* command is typed. The default is no.
- REPLY-INSERT: If yes, the *reply* command includes the original message automatically. The default is no.
- ✔ **SAVED-MESSAGES-FILE**: Defines a file that will automatically receive copies of all your outgoing messages. There is no default. If the directory is not specified, the mail-directory is assumed.
- ✔ **SEND-PROMPT**: Defines the string to be used as the send prompt. The default is *Send*>.
- SEND-VERBOSE: If yes, after entering *send* at the send prompt to mail your outgoing message, the recipients of the mail you have just sent will be displayed. The default is yes.
- SENDMAIL-BACKGROUND: If yes, MM won't wait for the sendmail program to finish running before continuing with MM commands. The default is yes.
- SENDMAIL-VERBOSE: If yes, display the mail sender program messages about mail delivery. The default is no.

- SPELLER: Defines the spell checker that is to be used with the *spell* command. The default is *ispell*.
- SUSPEND-ON-EXIT: If yes, when using the *exit* command to get out of MM, the process will be suspended. The default is yes.
- SUSPEND-ON-QUIT: If yes, when using the *quit* command to get out of MM, the process will be suspended. The default is yes.
- TEMP-DIRECTORY: Gives the directory to create temporary files in. The default is ~, your home directory.
- ✔ **TERSE-TEXT-PROMPT**: If yes, the prompt for message body in text mode will be just **Msg:**. The default is no.
- TOP-LEVEL-PROMPT: Gives the string to be used as the top-level command prompt. The default is *MM*>.
- USE-CRT-FILTER-ALWAYS: If yes, messages will be piped to crt-filter regardless of length. If no, your crt-filter will be used only when the message is too big to fit on your screen all at once. The default is no.
- ✔ **USE-EDITOR-ALWAYS**: If yes, the editor defined by the editor variable will automatically be used to compose messages. The default is no.
- USE-INVALID-ADDRESS: If yes, strange mail address formats will be accepted by MM in the *reply*, *forward*, and *edit* commands (e.g. hostname::username). The default is ask.
- USER-HEADERS: Gives a list of header fields you may wish to add to outgoing messages. Defining this variable allows for command completion to be used with the *user-header* command in *send* mode. There is no default.
- ✔ **USER-LEVEL**: Defines the level of expertise for the user. The options are *novice* and *expert*. If the level is set to novice, the one-line display of "hints" will appear at each of the prompts. At expert level, the "hints" disappear. The default is *novice*.
- USER-NAME: Defines the user name for a particular UID. For the general user population this command does not apply. If you would like to have a nickname appear in the *from* field, use the *personal-name* variable.

8. Special Topics

* Your Electronic Mail Address

An electronic mail address at Columbia looks like this one:

brennan@cunixf.cc.columbia.edu

In general, the format is *userid@host*, and the *host* portion is usually subdivided by periods.

The address shown above is interpreted as follows:

- *brennan* is the user ID. Some user IDs are the first or last name of the person. The most common form actually is initials, followed by a number to distinguish people with the same initials. Your user ID may therefore look like *jpb7*.
- *cunixf* is the machine where *brennan* is located. This is the "mail" host for officers and staff. Students are on cunixa, cunixb or cunixd.
- *cc* indicates the department responsible for the machine; *cc* stands for Computer Center. Some Columbia departments have their own machines, and will have a different designation here.
- *columbia.edu* identifies Columbia University.

Mail can be sent almost anywhere in the world by using a full address like the one shown above for *brennan*. There are numerous electronic mail networks in the United States and in the rest of the world that interconnect.

Your correspondents do not have to use MM as their mail program. Each mail program actually is translating mail into and out of a common format, making it possible for correspondents to be on very different types of machines that run different mail programs.

The electronic mail network has been patched together, and there are some exceptions to the above statements. You may find you want to correspond with someone whose address does not match the usual format or whose mail program does not work well with the common format that MM uses. For advice, call the Help Desk at 854-4854, or send mail to *consultant*, and experienced CUCCA staff will look into the problem.

One of the main obstacles to communication is that there are very few electronic mail directories. Usually, you and your colleague will need to exchange addresses by some other contact before you can start sending mail. Ask people to send a test message to your address.

Received mail will show the address it came from in the header (see the next page). In some cases, you also see a routing, a list of intermediate addresses through which the message passed. MM's *reply* command will send mail back to the address shown in the *from* field.

You might use the *define* command (page 22) to create an alias for someone you will mail to regularly. Full addresses are difficult to remember and to type accurately.

Mail that cannot be delivered is returned, after a time, with a system message indicating why delivery failed. Look at the header and see whether you typed the address correctly. If you don't see what the problem is, a good way to get help is to use MM's *forward* command to send the returned message to *consultant*, so CUCCA experts can examine the full header and the message about nondelivery, and advise on what else to try.

Within Columbia, mail can be addressed in short form. The system assumes the rest of the address is the same as the sender's. For example, another cunixf user can address just to *brennan*, and a user on cuvmb can address just to *brennan@cunixf*. (By the way, MM is not available on cuvmb, so the user there would be using some other mail program, like VMM or MAIL, but that will not be apparent to *brennan*.)

In the near future (as we go to press), the address *columbia.edu* will be set up so that outsiders do not have to specify where at Columbia (like *cunixf.cc*), but can just send to *userid@columbia.edu*, and mail will be directed by the system to the correct machine. The full address will still be good as well.

HEADERS

This is the header of a message sent from a person at Johns Hopkins University in reply to a message from *brennan* at Columbia:

```
Return-Path: <xyz@welchlab.welch.jhu.edu>
Received: from welchlab.welch.jhu.edu by cunixf.cc.columbia.edu (5.59/FCB)
        id AA03285; Fri, 29 Sep 89 12:47:40 EDT
Received: by welchlab.welch.jhu.edu (4.0/4.0)
        id AA29525; Fri, 29 Sep 89 12:47:22 EST
From: Archy Bargy <xyz@welchlab.welch.jhu.edu>
Message-Id: <8909291747.AA29525@welchlab.welch.jhu.edu>
Subject: Re: ELM...
To: brennan@cunixf.cc.columbia.edu (Joseph Brennan)
Date: Fri, 29 Sep 89 12:47:22 EST
In-Reply-To: <CMM.0.88.623089573.brennan@cunixf.cc.columbia.edu>; from "Joseph \
Brennan" at Sep 29, 89 12:26 (noon)
X-Mailer: Elm [version 2.1 PL1]
```

The *from* field, about halfway down, shows the user is Archy Bargy, and most importantly that his user ID is *xyz@welchlab.welch.jhu.edu*, which is the address to use to send him mail. The format is very similar to Columbia's. The first field of this message, *return-path*, also shows the correct address, but sometimes does not appear.

Headers on mail from outside Columbia can be much longer. Since most of the data is not of much interest, you may want to hide some of the headers when you read mail. One of the customization commands, *set dont-type-headers*, will hide whatever header fields you name as arguments. For example, if you customize with *set dont-type-headers received, return-path, message-ID, resent-message-ID*, then most of the excess material in the example would not appear on screen. If you do use *set*, remember to type *save-init* before leaving MM, to save the setting.

* Other Users: the Shell Command *finger*

The MM command *finger* actually does the shell command *finger*. You will get the same results whether you type the command at an MM prompt or at the shell prompt *$*. In these examples, the shell prompt is shown. Note the use of **small** and **CAPITAL** letters.

The command *finger* by itself shows who is logged in right now:

```
$ finger
Umax 4.3 (B4_0.16) XPC NFS Fri Nov  3 15:39:12 1989

login     name            tty idle when      location         type
brennan   Joseph Brennan   p3      Fri 08:23 128.59.35.52     us
caldano   Dan Caldano      A3      Fri 08:57 info01.cc.columb cul
dcl       Don Lanini       52      Fri 09:09 info05.cc.columb us
...
```

The display is usually so long that it runs off the screen. To hold it still, it is actually better to type ***finger | more***; the pipe symbol (|) sends the data into *more*, another command that displays data one screen at a time.

The command *finger -Q* with a name or part of a name may help find a user ID:

```
$ finger -Q dan
Dan Caldano                           caldano
Dan Cooper                            cooper
Daniel B. Dobkin                      dbd
Daniel J Lazarus                      djl
Danny M Lee                           dml
Alena P Danchak                       ptak
...
```

This display shows all users that exist whose first or last names start with "dan". The column on the right shows their user IDs.

The command *finger* with a user ID shows information about the user:

```
$ finger brennan
brennan    Joseph Brennan        p3        Fri 08:23 128.59.35.52  us
No new mail.
No plan.
```

In this case, Joe Brennan is logged in now. If he were not, in place of *p3 Fri 08:23...* would be *Last login <date, time>....* The phrase *No new mail* means no mail is waiting to be read. If you had sent Joe mail, you can infer from *no new mail* and from the login time that he’s read it by now. The command *finger -v* with a user ID shows a more "verbose" version.

For more on *finger*, type *man finger* at the shell prompt, or type *!man finger* at an MM prompt.

* Concluding Hints

COMMAND RETRY

After you become more adept at using MM, you may want to use a feature called *command retry*. This means that after you have typed a command and hit return, then realize that was not what you meant to do, you can redo the command. When you get the prompt back from MM, if the **very next** thing you type is <*ctrl/p*>, MM will redisplay your last command. You can then delete the offending parts and run the command you wanted. This saves you from having to retype the entire command. You can also use this command in conjunction with the command <*ctrl/n*> to go backwards and forwards (<*ctrl/n*> moves you forward) through the commands you have typed during the current MM session.

COMMENTS

MM will accept comments on any line. Comments are useful inside files run by the *take* command (see *help take*). Precede comments by *#*. If you want to put special characters like *#* or *[ESC]* into text, use the quoting character, *ctrl/v*. *ctrl/v* followed by any character simply inserts that character. Thus, *#* indicates that the rest of the line is comment, while *^v#* (*ctrl/v* and then *#*) indicates literally the character *#*.

FILES USED BY MM AT STARTUP

Whenever MM starts up, it first *take*s the file *~/.mminit* to set up your environment. Since reading of the *.mminit* file can be somewhat slow, MM knows how to write a faster version of it, called *~/.mmfast*. You will never need to modify this file in any way, since MM takes care of keeping it up to date with your *.mminit* file. *.mmfast* is a very small file, but if you feel you are terribly short on disk space, you can put *set fast-init-file no* in your *.mminit*, and MM will stop writing *.mmfast*. You can then delete it.

After your *.mminit* file is *take*n, MM then *take*s the file *~/.mmrc*, which contains other MM commands. For example, if you want to read your new mail automatically **every** time you go into MM, you can put the *read* command into your *.mmrc* file.

If you have a *.mailrc* file in your directory (a file with aliases for the "Mail" program), it is also *take*n upon entry to MM to find the aliases. Other "mail" commands are ignored.

See the *take* command for an explanation of how other files can be *take*n also.

SHELL COMMANDS

You can issue shell commands from inside MM. Just precede the command with the exclamation point "!".

BELLS

Type a <*ctrl/g*> into your text to insert a bell sound (actually a "beep" on most terminals). Since the bell can be annoying to readers, save it for appropriate times.

COMMUNICATIONS BREAKTHROUGH

One disadvantage of electronic mail is that your readers may misunderstand your tone of voice. You can state something ironically and be taken seriously. For suggestions on how to overcome the problem, type *man smiley* at the shell prompt.