R-2367-AF
November 1979

The MH Message Handling System: User's Manual

Bruce S. Borden, R. Stockton Gaines, Norman Z. Shapiro

A Project AIR FORCE report
prepared for the
United States Air Force

Rand
SANTA MONICA, CA. 90406

The research reported here was sponsored by the Directorate of Operational Requirements, Deputy Chief of Staff/Research, Development, and Acquisition, Hq USAF, under Contract F49620-77-C-0023. The United States Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon.

Library of Congress Cataloging in Publication Data

Borden, Bruce S 1951-
The MH message handling system.

([Report] - Rand Corporation ; R2367-AF)
"A Project Air Force report, prepared for the United States Air Force."
1. Telecommunication--Message processing. 2. Communications, Military. 3. United States--Armed Forces--Communication systems. I. Gaines, R. Stockton, 1934- joint author. II. Shapiro, Norman Zalmon, 1932- joint author. III. Title. IV. Series: Rand Corporation. Rand report ; R-2367-AF.
AS36.R3 R-2367 [UA943] 081s [623.7'3] 79-22419
ISBN 0-8330-0168-X

Published by The Rand Corporation

R-2367-AF
November 1979

The MH Message Handling System: User's Manual

Bruce S. Borden, R. Stockton Gaines, Norman Z. Shapiro

A Project AIR FORCE report
prepared for the
United States Air Force

Rand
SANTA MONICA, CA. 90406

PREFACE

This report describes a system for dealing with messages transmitted on a computer. Such messages might originate with other users of the same computer or might come from an outside source through a network to which the user's computer is connected. Such computer-based message systems are becoming increasingly widely used, both within and outside the Department of Defense.

The message handling system MH was developed for two reasons. One was to investigate some research ideas concerning how a message system might take advantage of the architecture of the UNIX time-sharing operating system for Digital Equipment Corporation PDP-11 and VAX computers, and the special features of UNIX's command-level interface with the user (the "shell"). The other reason was to provide a better and more adaptable base than that of conventional designs on which to build a command and control message system. The effort has succeeded in both regards, although this report mainly describes the message system itself and how it fits in with UNIX. The main research results are being described and analyzed in a forthcoming Rand report. The system is currently being used as part of a tactical command and control "laboratory", which is also being described in a separate report.

The present report should be of interest to three groups of readers. First, it is a complete reference manual for the users of MH (although users outside of Rand must take into account differences from the local Rand operating system). Second, it should be of interest to those who have a general knowledge of computer-based message systems, both in civilian and military applications. Finally, it should be of interest to those who build large subsystems that interface with users, since it illustrates a new approach to such interfaces.

The MH system was developed by the first author, using an approach suggested by the other two authors. Valuable assistance was provided by Phyllis Kantar in the later stages of the system's implementation. Several colleagues contributed to the ideas included in this system, particularly Robert Anderson and David Crocker. In addition, valuable experience in message systems, and a valuable source of ideas, was available to us in the form of a previous message system for UNIX called MS, designed at Rand by David Crocker.

This report was prepared as part of the Rand project entitled "Data Automation Research", sponsored by Project AIR FORCE.

SUMMARY

Electronic communication of text messages is becoming commonplace. Computer-based message systems—software packages that provide tools for dealing with messages—are used in many contexts. In particular, message systems are becoming increasingly important in command and control and intelligence applications.

This report describes a message handling system called MH. This system provides the user with tools to compose, send, receive, store, retrieve, forward, and reply to messages. MH has been built on the UNIX time-sharing system, a popular operating system developed for the DEC PDP-11 and VAX classes of computers.

A complete description of MH is given for users of the system. For those who do not intend to use the system, this description gives a general idea of what a message system is like. The system involves some new ideas about how large subsystems can be constructed. These design concepts and a comparison of MH with other message systems will be published in a forthcoming Rand report.

The interesting and unusual features of MH include the following: The user command interface to MH is the UNIX "shell" (the standard UNIX command interpreter). Each separable component of message handling, such as message composition or message display, is a separate command. Each program is driven from and updates a private user environment, which is stored as a file between program invocations. This private environment also contains information to "custom tailor" MH to the individual's tastes. MH stores each message as a separate file under UNIX, and it utilizes the tree-structured UNIX file system to organize groups of files within separate directories or "folders." All of the UNIX facilities for dealing with files and directories, such as renaming, copying, deleting, cataloging, off-line printing, etc., are applicable to messages and directories of messages (folders). Thus, important capabilities needed in a message system are available in MH without the need (often seen in other message systems) for code that duplicates the facilities of the supporting operating system. It also allows users familiar with the shell to use MH with minimal effort.

CONTENTS

1. INTRODUCTION

Although people can travel cross-country in hours and can reach others by telephone in seconds, communications still depend heavily upon paper, most of which is distributed through the mails.

There are several major reasons for this continued dependence on written documents. First, a written document may be proofread and corrected prior to its distribution, giving the author complete control over his words. Thus, a written document is better than a telephone conversation in this respect. Second, a carefully written document is far less likely to be misinterpreted or poorly translated than a phone conversation. Third, a signature offers reasonable verification of authorship, which cannot be provided with media such as telegrams.

However, the need for fast, accurate, and reproducible document distribution is obvious. One solution in widespread use is the telefax. Another that is rapidly gaining popularity is electronic mail. Electronic mail is similar to telefax in that the data to be sent are digitized, transmitted via phone lines, and turned back into a document at the receiver. The advantage of electronic mail is in its compression factor. Whereas a telefax must scan a page in very fine lines and send all of the black and white information, electronic mail assigns characters fixed codes which can be transmitted as a few bits of information. Telefax presently has the advantage of being able to transmit an arbitrary page, including pictures, but electronic mail is beginning to deal with this problem. Electronic mail also integrates well with current directions in office automation, allowing documents prepared with sophisticated equipment at one site to be quickly transferred and printed at another site.

Currently, most electronic mail is intraorganizational, with mail transfer remaining within one computer. As computer networking becomes more common, however, it is becoming more feasible to communicate with anyone whose computer can be linked to your own via a network.

The pioneering efforts on general-purpose electronic mail were by organizations using the Defense Department's ARPANET.[1] The capability to send messages between computers existed before the ARPANET was developed, but it was used only in limited ways. With the advent of the ARPANET, tools began to be developed which made it convenient for individuals or organizations to distribute messages over broad geographic areas, using diverse computer facilities. The interest and activity in message systems has now reached such proportions that steps have been taken within the DoD to coordinate and unify the development of military message systems. The use of electronic mail is expected to increase dramatically in the next few years. The utility of such systems in the command and control and intelligence environments is clear, and applications in these areas will probably lead the way. As the costs for sending and handling electronic messags continue their rapid decrease, such uses can be expected to spread rapidly into other areas and, of course, will not be limited to the DoD.

A message system provides tools that help users (individuals or organizations) deal with messages in various ways. Messages must be composed, sent, received, stored, retrieved, forwarded, and replied to. Today's best interactive computer systems provide a variety of word-processing and information handling capabilities. The message handling facilities should be well integrated with the rest of the system, so as to be a graceful extension of overall system capability.

The message system described in this report, MH, provides most of the features that can be found in other message systems and also incorporates some new

ones. It has been built on the UNIX time-sharing system,[2] a popular operating system for the DEC PDP-11 and VAX classes of computers. A "secure" operating system similar to UNIX is currently being developed,[3] and that system will also run MH.

This report provides a complete description of MH and thus may serve as a user's manual, although parts of the report will be of interest to non-users as well. Sections 2 and 3, the Overview and Tutorial, present the key ideas of MH and will give those not familiar with message systems an idea of what such systems are like.

MH consists of a set of commands which use some special files and conventions. Section 4 covers the information a user needs to know in addition to the commands. The final section, Sec. 5, describes each of the MH commands in detail. A summary of the commands is given in Appendix A, and Appendixes B and C describe the ARPANET conventions for messages (we expect that many users of MH will be using the ARPANET) and the formal syntax of such messages, respectively. Finally, Appendix D provides an illustration of how MH commands may be used in conjunction with other UNIX facilities.

A novel approach has been taken in the design of MH. The design concept will be reported in detail in a forthcoming Rand report, but it can be described briefly as follows. Instead of creating a large subsystem that appears as a single command to the user (such as MS[4]) MH is a collection of separate commands which are run as separate programs. The file and directory system of UNIX are used directly. Messages are stored as individual files (datasets), and collections of them are grouped into directories. In contrast, most other message systems store messages in a complicated data structure within a monolithic file. With the MH approach, UNIX commands can be interleaved with commands invoking the functions of the message handler. Conversely, existing UNIX commands can be used in connection with messages. For example, all the usual UNIX editing, text-formatting, and printing facilities can be applied directly to individual messages. MH, therefore, consists of a relatively small amount of new code; it makes extensive use of other UNIX software to provide the capabilities found in other message systems.

2. OVERVIEW

There are three main aspects of MH: the way messages are stored (the message database), the user's profile (which directs how certain actions of the message handler take place), and the commands for dealing with messages.

Under MH, each message is stored as a separate file. A user can take any action with a message that he could with an ordinary file in UNIX. A UNIX directory in which messages are stored is called a folder. Each folder contains some standard entries to support the message-handling functions. The messages in a folder have numerical names. These folders (directories) are entries in a particular directory path, described in the user profile, through which MH can find message folders. Using the UNIX "link" facility, it is possible for one copy of a message to be "filed" in more than one folder, providing a message index facility. Also, using the UNIX tree-structured file system, it is possible to have a folder within a folder. This two-level organization provides a "selection-list" facility, with the full power of the MH commands available on selected sublists of messages.

Each user of MH has a user profile, a file in his $HOME (initial login) directory called ". mh_profile". This profile contains several pieces of information used by the MH commands: a path name to the directory that contains the message folders, information concerning which folder the user last referenced (the "current" folder), and parameters that tailor MH commands to the individual user's requirements. It also contains most of the necessary state information concerning how the user is dealing with his messages, enabling MH to be implemented as a set of individual UNIX commands, in contrast to the usual approach of a monolithic subsystem.

In MH, incoming mail is appended to the end of a file called . mail in a user's $HOME directory. The user adds the new messages to his collection of MH messages by invoking the command *inc*. *Inc* (incorporate) adds the new messages to a folder called "inbox", assigning them names which are consecutive integers starting with the next highest integer available in inbox. *Inc* also produces a *scan* summary of the messages thus incorporated.

There are four commands for examining the messages in a folder: *show*, *prev*, *next*, and *scan*. *Show* displays a message in a folder, *prev* displays the message preceding the current message, and *next* displays the message following the current message. *Scan* summarizes the messages in a folder, producing one line per message, showing who the message is from, the date, the subject, etc.

The user may move a message from one folder to another with the command *file*. Messages may be removed from a folder by means of the command *rmm*. In addition, a user may query what the current folder is and may specify that a new folder become the current folder, through the command *folder*.

A set of messages based on content may be selected by use of the command *pick*. This command searches through messages in a folder and selects those that match a given criterion. A subfolder is created within the original folder, containing links to all the messages that satisfy the selection criteria.

A message folder (or subfolder) may be removed by means of the command *rmf*.

There are five commands enabling the user to create new messages and send them: *comp*, *dist*, *forw*, *repl*, and *send*. *Comp* provides the facility for the user to compose a new message; *dist* redistributes mail to additional addressees; *forw* enables the

user to forward messages; and *repl* facilitates the generation of a reply to an incoming message. If a message is not sent directly by one of these commands, it may be sent at a later time using the command *send*.

All of the elements summarized above are described in more detail in the following sections. Many of the normal facilities of UNIX provide additional capabilities for dealing with messages in various ways. For example, it is possible to print messages on the line-printer without requiring any additional code within MH. Using standard UNIX facilities, any terminal output can be redirected to a file for repeated or future viewing. In general, the flexibility and capabilities of the UNIX interface with the user are preserved as a result of the integration of MH into the UNIX structure.

3. TUTORIAL

This tutorial provides a brief introduction to the MH commands. It should be sufficient to allow the user to read his mail, do some simple manipulations of it, and create and send messages.

A message has two major pieces: the header and the body. The body consists of the text of the message (whatever you care to type in). It follows the header and is separated from it by an empty line. (When you compose a message, the form that appears on your terminal shows a line of dashes after the header. This is for convenience and is replaced by an empty line when the message is sent.) The header is composed of several components, including the subject of the message and the person to whom it is addressed. Each component starts with a name and a colon; components must not start with a blank. The text of the component may take more than one line, but each continuation line must start with a blank. Messages typically have "to:", "cc:", and "subject:" components. When composing a message, you should include the "to:" and "subject:" components; the "cc:" (for people you want to send copies to) is not necessary.

The basic MH commands are *inc*, *scan*, *show*, *next*, *prev*, *rmm*, *comp*, and *repl*. These are described below.

inc

When you get the message "You have mail", type the command *inc*. You will get a "scan listing" such as:

```
7+    7/13    Cas          revival of measurement work
8     10/ 9   Norm         NBS people and publications
9     11/26   To:norm       question ≪Are there any functions
```

This shows the messages you received since the last time you executed this command (*inc* adds these new messages to your inbox folder). You can see this list again, plus a list of any other messages you have, by using the *scan* command.

scan

The scan listing shows the message number, followed by the date and the sender. (If you are the sender, the addressee in the "to:" component is displayed. You may send yourself a message by including your name among the "to:" or "cc:" addressees.) It also shows the message's subject; if the subject is short, the first part of the body of the message is included after the characters ≪.

show

This command shows the current message, that is, the first one of the new messages after an *inc*. If the message is not specified by name (number), it is generally the last message referred to by an MH command. For example,

show 5 will show message 5.

You can use the show command to copy a message or print a message.

show > x	will copy the message to file x.
show | print	will print the message, using the *print* command.
next	will show the message that follows the current message.
prev	will show the message previous to the current message.
rmm	will remove the current message.
rmm 3	will remove message 3.

comp

The *comp* command puts you in the editor to write or edit a message. Fill in or delete the "to:", "cc:", and "subject:" fields, as appropriate, and type the body of the message. Then exit normally from the editor. You will be asked "What now?". Type a carriage return to see the options. Typing **send** will cause the message to be sent; typing **quit** will cause an exit from *comp*, with the message draft saved.

If you quit without sending the message, it will be saved in a file called /usr/<name>/Mail/draft (where /usr/<name> is your $HOME directory). You can edit this file and send the message later, using the *send* command.

comp −editor prompter

This command uses a different editor and is useful for preparing "quick and dirty" messages. It prompts you for each component of the header. Type the information for that component, or type a carriage return to omit the component. After that, type the body of the message. Backspacing is the only form of editing allowed with this editor. When the body is complete, type a carriage return followed by <CTRL-D> (<OPEN> on Ann Arbor terminals). This completes the initial preparation of the message; from then on, use the same procedures as with *comp* (above).

repl
repl n

This command makes up an initial message form with a header that is appropriate for replying to an existing message. The message being answered is the current message if no message number is mentioned, or n if a number is specified. After the header is completed, you can finish the message as in *comp* (above).

This is enough information to get you going using MH.[10] There are more commands, and the commands described here have more features. Subsequent sections explain MH in complete detail. The system is quite powerful if you want to use its sophisticated features, but the foregoing commands suffice for sending and receiving messages.

There are numerous additional capabilities you may wish to explore. For example, the *pick* command will select a subset of messages based on specified criteria such as sender or subject. Groups of messages may be designated, as described in Sec. V, under "Message Naming". The file ". mh_profile" can be used to tailor your use of the message system to your needs and preferences, as described in Sec. V, under "The User Profile". In general, you may learn additional features of the system selectively, according to your requirements, by studying the relevant sections of this manual. There is no need to learn all the details of the system at once.

4. DETAILED DESCRIPTION

This section describes the MH system in detail, including the components of the user profile, the conventions for message naming, and some of the other MH conventions. Readers who are generally familiar with computer systems will be able to follow the principal ideas, although some details may be meaningful only to those familiar with UNIX.

THE USER PROFILE

The first time an MH command is issued by a new user, the system prompts for a "path" and creates an MH "profile".

Each MH user has a profile which contains current state information for the MH package and, optionally, tailoring information for each individual program. When a folder becomes the current folder, it is recorded in the user's profile. Other profile entries control the MH path (where folders and special files are kept), folder and message protections, editor selection, and default arguments for each MH program.

The MH profile is stored in the file ". mh_profile" in the user's $HOME directory. It has the format of a message without any body. That is, each profile entry is on one line, with a keyword followed by a colon (:) followed by text particular to the keyword.

☛ *This file must not have blank lines.*

The keywords may have any combination of upper and lower case. (See Appendix B for a description of message formats.)

For the average MH user, the only profile entry of importance is "Path". Path specifies a directory in which MH folders and certain files such as "draft" are found. The argument to this keyword must be a legal UNIX path that names an existing directory. If this path is unrooted (i.e., does not begin with a /), it will be presumed to start from the user's $HOME directory. All folder and message references within MH will relate to this path unless full path names are used.

Message protection defaults to 664, and folder protection to 751. These may be changed by profile entries "Msg-Protect" and "Folder-Protect", respectively. The argument to these keywords is an octal number which is used as the UNIX file mode.[1]

When an MH program starts running, it looks through the user's profile for an entry with a keyword matching the program's name. For example, when *comp* is run, it looks for a "comp" profile entry. If one is found, the text of the profile entry is used as the default switch setting until all defaults are overridden by explicit switches passed to the program as arguments. Thus the profile entry "comp: −form standard.list" would direct *comp* to use the file "standard.list" as the message skeleton. If an explicit form switch is given to the *comp* command, it will override the switch obtained from the profile.

In UNIX, a program may exist under several names, either by linking or aliasing. The actual invocation name is used by an MH program when scanning for its profile defaults. Thus, each MH program may have several names by which it can be

[1] See *chmod*(1) in the *UNIX Programmer's Manual*.[5]

invoked, and each name may have a different set of default switches. For example, if *comp* is invoked by the name *icomp*, the profile entry "icomp" will control the default switches for this invocation of the *comp* program. This provides a powerful definitional facility for commonly used switch settings.

The default editor for editing within *comp*, *repl*, *forw*, and *dist*, is "/bin/ned".[2] A different editor may be used by specifying the profile entry "Editor: ". The argument to "Editor" is the name of an executable program or shell command file which can be found via the user's $PATH defined search path, excluding the current directory. The "Editor:" profile specification may in turn be overridden by a "−editor <editor>" profile switch associated with *comp*, *repl*, *forw*, or *dist*. Finally, an explicit editor switch specified with any of these four commands will have ultimate precedence.

During message composition, more than one editor may be used. For example, one editor (such as *prompter*) may be used initially, and a second editor may be invoked later to revise the message being composed (see the discussion of *comp* in Section 5 for details). A profile entry "<lasteditor>−next: <editor>" specifies the name of the editor to be used after a particular editor. Thus "comp: −e prompter" causes the initial text to be collected by *prompter*, and the profile entry "prompter−next: ed" names ed as the editor to be invoked for the next round of editing.

Some of the MH commands, such as *show*, can be used on message folders owned by others, if those folders are readable. However, you cannot write in someone else's folder. All the MH command actions not requiring write permission may be used with a "read-only" folder. In a writable folder, a file named "cur" is used to contain its current message name. For read-only folders, the current message name is stored in the user's profile.

Table 1 lists examples of the currently defined profile entries, typical arguments, and the programs that reference the entries.

Table 1

PROFILE COMPONENTS

Keyword and Argument	MH Programs that Use Component
Path: Mail	All
Current-Folder: inbox	Most
Editor: /bin/ed	*comp, dist, forw, repl*
Msg−Protect: 644	*inc*
Folder−Protect: 711	*file, inc, pick*
<program>: default switches	All
cur−<read-onlyfolder>: 172	Most
prompter−next: ed	*comp, dist, forw, repl*

Path should be present. Folder is maintained automatically by many MH commands (see the "Context" sections of the individual commands in Sec. V). All

[2]See Ref. 6 for a description of the NED text editor.

other entries are optional, defaulting to the values described above.

MESSAGE NAMING

Messages may be referred to explicitly or implicitly when using MH commands. A formal syntax of message names is given in Appendix C, but the following description should be sufficient for most MH users. Some details of message naming that apply only to certain commands are included in the description of those commands.

Most of the MH commands accept arguments specifying one or more folders, and one or more messages to operate on. The use of the word "msg" as an argument to a command means that exactly one message name may be specified. A message name may be a number, such as 1, 33, or 234, or it may be one of the "reserved" message names: first, last, prev, next, and cur. (As a shorthand, a period (.) is equivalent to cur.) The meanings of these names are straightforward: "first" is the first message in the folder; "last" is the last message in the folder; "prev" is the message numerically previous to the current message; "next" is the message numerically following the current message; "cur" (or ". ") is the current message in the folder.

The default in commands that take a "msg" argument is always "cur".

The word "msgs" indicates that several messages may be specified. Such a specification consists of several message designations separated by spaces. A message designation is either a message name or a message range. A message range is a specification of the form name1−name2 or name1:n, where name1 and name2 are message names and n is an integer. The first form designates all the messages from name1 to name2 inclusive; this must be a non-empty range. The second form specifies up to n messages, starting with name1 if name1 is a number, or first, cur, or next, and ending with name1 if name1 is last or prev. This interpretation of n is overridden if n is preceded by a plus sign or a minus sign; +n always means up to n messages starting with name1, and −n always means up to n messages ending with name1. Repeated specifications of the same message have the same effect as a single specification of the message. Examples of specifications are:

```
1 5 7−11 22
first 6 8 next
first−10
last:5
```

The message name "all" is a shorthand for "first−last", indicating all of the messages in the folder.

The limit on the number of messages in an expanded message list is generally 999−the maximum number of messages in a folder. However, the *show* command and the commands '*pick* −*scan*' and '*pick* −*show*' are constrained to have argument lists that are no more than 512 characters long. (Under Version 7 UNIX this limit is 4096.)

In commands that accept "msgs" arguments, the default is either cur or all, depending on which makes more sense.

In all of the MH commands, a plus sign preceding an argument indicates a folder name. Thus, "+inbox" is the name of the user's standard inbox. If an explicit folder argument is given to an MH command, it will become the current folder (that is, the "Current-Folder:" entry in ". mh_profile" will be changed to this

folder). In the case of the *file* and *pick* commands, which can have multiple output folders, a new source folder (other than the default current folder) is specified by "−src +folder".

OTHER MH CONVENTIONS

One very powerful feature of MH is that the MH commands may be issued from any current directory, and the proper path to the appropriate folder(s) will be taken from the user's profile. If the MH path is not appropriate for a specific folder or file, the automatic prepending of the MH path can be avoided by beginning a folder or file name with /. Thus any specific full path may be specified.

Arguments to the various programs may be given in any order, with the exception of a few switches whose arguments must follow immediately, such as "−src +folder" for *pick* and *file*.

Whenever an MH command prompts the user, the valid options will be listed in response to a <RETURN>. (The first of the listed options is the default if end-of-file is encountered, such as from a command file.) A valid response is any *unique* abbreviation of one of the listed options.

Standard UNIX documentation conventions are used in this report to describe MH command syntax. Arguments enclosed in brackets ([ ]) are optional; exactly one of the arguments enclosed within braces ({ }) must be specified, and all other arguments are required. The use of ellipsis dots (...) indicates zero or more repetitions of the previous item. For example, "+folder ..." would indicate that one or more "+folder" arguments is required and "[+folder ...]" indicates that 0 or more "+folder" arguments may be given.

MH departs from UNIX standards by using switches that consist of more than one character, e.g. "−header". To minimize typing, only a unique abbreviation of a switch need be typed; thus, for "−header", "−hea" is probably sufficient, depending on the other switches the command accepts. Each MH program accepts the switch "−help" (which *must* be spelled out fully) and produces a syntax description and a list of switches. In the list of switches, parentheses indicate required characters. For example, all "−help" switches will appear as "−(help)", indicating that no abbreviation is accepted.

Many MH switches have both on and off forms, such as "−format" and "−noformat". In many of the descriptions in Sec. V, only one form is defined; the other form, often used to nullify profile switch settings, is assumed to be the opposite.

MH COMMANDS

The MH package comprises 16 programs:

comp	Compose a message
dist	Redistribute a message
file	Move messages between folders
folder	Select/list status of folders
forw	Forward a message
inc	Incorporate new mail
next	Show the next message
pick	Select a set of messages by context
prev	Show the previous message
prompter	Prompting editor front end for composing messages
repl	Reply to a message
rmf	Remove a folder
rmm	Remove messages
scan	Produce a scan listing of selected messages
send	Send a previously composed message
show	Show messages

These programs are described below. The form of the descriptions conforms to the standard form for the description of UNIX commands.

NAME

comp − compose a message

SYNOPSIS

comp [−editor editor] [−form formfile] [file] [−use] [−nouse] [−help]

DESCRIPTION

Comp is used to create a new message to be mailed. If *file* is not specified, the file named "draft" in the user's MH directory will be used. *Comp* copies a message form to the file being composed and then invokes an editor on the file. The default editor is /bin/ned, which may be overridden with the '−editor' switch or with a profile entry "Editor:". (See Ref. 5 for a description of the NED text editing system.) The default message form contains the following elements:

```
To:
cc:
Subject:
---------
```

If the file named "components" exists in the user's MH directory, it will be used instead of this form. If '−form formfile' is specified, the specified formfile (from the MH directory) will be used as the skeleton. The line of dashes or a blank line must be left between the header and the body of the message for the message to be identified properly when it is sent (see *send*;). The switch '−use' directs *comp* to continue editing an already started message. That is, if a *comp* (or *dist*, *repl*, or *forw*) is terminated without sending the message, the message can be edited again via "comp −use".

If the specified file (or draft) already exists, *comp* will ask if you want to delete it before continuing. A reply of **No** will abort the *comp*, **yes** will replace the existing draft with a blank skeleton, **list** will display the draft, and **use** will use it for further composition.

Upon exiting from the editor, *comp* will ask "What now?". The valid responses are **list**, to list the draft on the terminal; **quit**, to terminate the session and preserve the draft; **quit delete**, to terminate, then delete the draft; **send**, to send the message; **send verbose**, to cause the delivery process to be monitored; **edit <editor>**, to invoke <editor> for further editing; and **edit**, to re-edit using the same editor that was used on the preceding round unless a profile entry "<lasteditor> −next: <editor>" names an alternative editor.

Files

/etc/mh/components	The message skeleton
or <mh-dir>/components	Rather than the standard skeleton
$HOME/. mh_profile	The user profile
<mh-dir>/draft	The default message file
/usr/bin/send	To send the composed message

Profile Components

Path:	To determine the user's MH directory
Editor:	To override the use of /bin/ned as the default editor
<lasteditor> −next:	To name an editor to be used after exit from <lasteditor>

Defaults

'file' defaults to draft
'−editor' defaults to /bin/ned
'−nouse'

Context

Comp does not affect either the current folder or the current message.

NAME

dist − redistribute a message to additional addresses

SYNOPSIS

dist [+folder] [msg] [−form formfile] [−editor editor] [−annotate] [−noannotate] [−inplace] [−noinplace] [−help]

DESCRIPTION

Dist is similar to *forw*. It prepares the specified message for redistribution to addresses that (presumably) are not on the original address list. The file "distcomps" in the user's MH directory, or a standard form, or the file specified by '−form formfile' will be used as the blank components file to be prepended to the message being distributed. The standard form has the components "Distribute-to:" and "Distribute-cc:". When the message is sent, "Distribution-Date: date", "Distribution-From: name", and "Distribution-Id: id" (if '−msgid' is specified to *send*,) will be prepended to the outgoing message. Only those addresses in "Distribute-To", "Distribute-cc", and "Distribute-Bcc" will be sent. Also, a "Distribute-Fcc: folder" will be honored (see *send;*).

Send recognizes a message as a redistribution message by the existence of the field "Distribute-To:", so don't try to redistribute a message with only a "Distribute-cc:".

If the '−annotate' switch is given, each message being distributed will be annotated with the lines:

Distributed: ≪date≫
Distributed: Distribute-to: names

where each "to" list contains as many lines as required. This annotation will be done only if the message is sent directly from *dist*. If the message is not sent immediately from *dist* (i.e., if it is sent later via *send;*), "comp −use" may be used to re-edit and send the constructed message, but the annotations won't take place. The '−inplace' switch causes annotation to be done in place in order to preserve links to the annotated message.

See *comp* for a description of the '−editor' switch and for options upon exiting from the editor.

Files

/etc/mh/components	The message skeleton
or <mh-dir>/components	Rather than the standard skeleton
$HOME/. mh_profile	The user profile
<mh-dir>/draft	The default message file
/usr/bin/send	To send the composed message

Profile Components

Path:	To determine the user's MH directory
Editor:	To override the use of /bin/ned as the default editor
<lasteditor>−next:	To name an editor to be used after exit from <lasteditor>

Defaults

'+folder' defaults to the current folder
'msg' defaults to cur
'−editor' defaults to /bin/ned
'−noannotate'
'−noinplace'

Context

If a +folder is specified, it will become the current folder, and the current message will be set to the message being redistributed.

NAME

file − file message(s) in (an)other folder(s)

SYNOPSIS

file [−src +folder] [msgs] [−link] [−preserve] +folder ... [−nolink] [−nopreserve] [−file file] [−nofile] [−help]

DESCRIPTION

File moves (*mv*(I)) or links (*ln*(I)) messages from a source folder into one or more destination folders. If you think of a message as a sheet of paper, this operation is not unlike filing the sheet of paper (or copies) in file cabinet folders. When a message is filed, it is linked into the destination folder(s) if possible, and is copied otherwise. As long as the destination folders are all on the same file system, multiple filing causes little storage overhead. This facility provides a good way to cross-file or multiply-index messages. For example, if a message is received from Jones about the ARPA Map Project, the command

file cur +jones +Map

would allow the message to be found in either of the two folders 'jones' or 'Map'.

The option '−file file' directs *file* to use the specified file as the source message to be filed, rather than a message from a folder.

If a destination folder doesn't exist, *file* will ask if you want to create one. A negative response will abort the file operation.

'−link' preserves the source folder copy of the message (i.e., it does a *ln*(I) rather than a *mv*(I)), whereas, '−nolink' deletes the "filed" messages from the source folder. Normally, when a message is filed, it is assigned the next highest number available in each of the destination folders. Use of the '−preserve' switch will override this message "renaming", but name conflicts may occur, so use this switch cautiously. (See *pick* for more details on message numbering.)

If '−link' is not specified (or '−nolink' is specified), the filed messages will be removed (unlink(II)) from the source folder.

Files

$HOME/. mh_profile	The user profile

Profile Components

Path:	To determine the user's MH directory
Current-Folder:	To find the default current folder
Folder−Protect:	To set mode when creating a new folder

Defaults

'−src +folder' defaults to the current folder
'msgs' defaults to cur
'−nolink'
'−nopreserve'
'−nofile'

Context

If '−src +folder' is given, it will become the current folder for future MH commands. If neither '−link' nor 'all' are specified, the current message in the source folder will be set to the last message specified; otherwise, the current message won't be changed.

NAME

folder − set/list current folder/message

SYNOPSIS

folder [+folder] [msg] [−all] [−fast] [−nofast] [−up] [−down] [−header] [−noheader] [−total] [−nototal] [−pack] [−nopack] [−help]

folders <equivalent to 'folder −all'>

DESCRIPTION

Since the MH environment is the shell, it is easy to lose track of the current folder from day to day. *Folder* will list the current folder, the number of messages in it, the range of the messages (low-high), and the current message within the folder, and will flag a selection list or extra files if they exist. An example of the output is:

inbox+ has 16 messages (3− 22); cur= 5.

If a '+folder' and/or 'msg' are specified, they will become the current folder and/or message. An '−all' switch will produce a line for each folder in the user's MH directory, sorted alphabetically. These folders are preceded by the read-only folders, which occur as . mh_profile "cur−" entries. For example,

```
                Folder      # of messages (  range  ); cur msg (other files)
      /fsd/rs/m/tacc  has    35 messages (  1− 35); cur= 23.
 /rnd/phyl/Mail/EP  has    82 messages (  1−108); cur= 82.
                  ff  has     4 messages (  1−  4); cur=  1.
             inbox+  has    16 messages (  3− 22); cur=  5.
                  mh  has    76 messages (  1− 76); cur= 70.
               notes  has     2 messages (  1−  2); cur=  1.
                ucom  has   124 messages (  1−124); cur=  6; (select).

                TOTAL= 339 messages in  7 Folders.
```

The "+" after inbox indicates that it is the current folder. The "(select)" indicates that the folder ucom has a selection list produced by *pick*. If "others" had appeared in parentheses at the right of a line, it would indicate that there are files in the folder directory that don't belong under the MH file naming scheme.

The header is output if either an '−all' or a '−header' switch is specified; it is suppressed by '−noheader'. Also, if *folder* is invoked by a name ending with "s" (e.g., *folders*), '−all' is assumed. A '−total' switch will produce only the summary line.

If '−fast' is given, only the folder name (or names in the case of '−all') will be listed. (This is faster because the folders need not be read.)

The switches '−up' and '−down' change the folder to be the one above or below the current folder. That is, "folder −down" will set the folder to "<current−folder>/select", and if the current folder is a selection-list folder, "folder −up" will set the current folder to the parent of the selection-list. (See *pick* for details on selection-lists.)

The '−pack' switch will compress the message names in a folder, removing holes in message numbering.

Files

$HOME/.mh_profile	The user profile
/bin/ls	To fast-list the folders

Profile Components

Path:	To determine the user's MH directory
Current-Folder:	To find the default current folder

Defaults

'+folder' defaults to the current folder
'msg' defaults to none
'-nofast'
'-noheader'
'-nototal'
'-nopack'

Context

If '+folder' and/or 'msg' are given, they will become the current folder and/or message.

NAME

forw − forward messages

SYNOPSIS

forw [+folder] [msgs] [−editor editor] [−form formfile] [−annotate] [−noannotate] [−inplace] [−noinplace] [−help]

DESCRIPTION

Forw may be used to prepare a message containing other messages. It constructs the new message from the components file or '−form formfile' (see *comp*), with a body composed of the message(s) to be forwarded. An editor is invoked as in *comp*, and after editing is complete, the user is prompted before the message is sent.

If the '−annotate' switch is given, each message being forwarded will be annotated with the lines

Forwarded: ≪date≫
Forwarded: To: names
Forwarded: cc: names

where each "To:" and "cc:" list contains as many lines as required. This annotation will be done only if the message is sent directly from *forw*. If the message is not sent immediately from *forw*, "comp −use" may be used in a later session to re-edit and send the constructed message, but the annotations won't take place. The '−inplace' switch permits annotating a message in place in order to preserve its links.

See *comp* for a description of the '−editor' switch.

Files

/etc/mh/components	The message skeleton
or <mh-dir>/components	Rather than the standard skeleton
$HOME/.mh_profile	The user profile
<mh-dir>/draft	The default message file
/usr/bin/send	To send the composed message

Profile Components

Path:	To determine the user's MH directory
Editor:	To override the use of /bin/ned as the default editor
Current-Folder:	To find the default current folder
<lasteditor>−next:	To name an editor to be used after exit from <lasteditor>

Defaults

'+folder' defaults to the current folder
'msgs' defaults to cur
'−editor' defaults to /bin/ned
'−noannotate'
'−noinplace'

Context

If a +folder is specified, it will become the current folder, and the current message will be set to the first message being forwarded.

NAME

inc − incorporate new mail

SYNOPSIS

inc [+folder] [−audit audit-file] [−help]

DESCRIPTION

Inc incorporates mail from the user's incoming mail drop (. mail) into an MH folder. If '+folder' isn't specified, the folder named "inbox" in the user's MH directory will be used. The new messages being incorporated are assigned numbers starting with the next highest number in the folder. If the specified (or default) folder doesn't exist, the user will be queried prior to its creation. As the messages are processed, a *scan* listing of the new mail is produced.

If the user's profile contains a "Msg−Protect: nnn" entry, it will be used as the protection on the newly created messages, otherwise the MH default of 664 will be used. During all operations on messages, this initially assigned protection will be preserved for each message, so *chmod*(I) may be used to set a protection on an individual message, and its protection will be preserved thereafter.

If the switch '−audit audit-file' is specified (usually as a default switch in the profile), then *inc* will append a header line and a line per message to the end of the specified audit-file with the format:

≪inc≫ date
<scan line for first message>
<scan line for second message>
<etc.>

This is useful for keeping track of volume and source of incoming mail. Eventually, *repl*, *forw*, *comp*, and *dist* may also produce audits to this (or another) file, perhaps with "Message-Id:" information to keep an exact correspondence history. "Audit-file" will be in the user's MH directory unless a full path is specified.

Inc will incorporate even illegally formatted messages into the user's MH folder, inserting a blank line prior to the offending component and printing a comment identifying the bad message.

In all cases, the . mail file will be zeroed.

Files

$HOME/. mh_profile	The user profile
$HOME/. mail	The user's mail drop
<mh-dir>/audit-file	Audit trace file (optional)

Profile Components

Path:	To determine the user's MH directory
Folder−Protect:	For protection on new folders
Msg−Protect:	For protection on new messages

Defaults

'+folder' defaults to "inbox"

Context

The folder into which the message is being incorporated will become the current folder, and the first message incorporated will be the current message. This leaves the context ready for a *show* of the first new message.

NAME

next − show the next message

SYNOPSIS

next [+folder] [−switches for *l*] [−help]

DESCRIPTION

Next performs a *show* on the next message in the specified (or current) folder. Like *show*, it passes any switches on to the program *l*, which is called to list the message. This command is exactly equivalent to "show next".

Files

$HOME/. mh_profile	The user profile

Profile Components

Path:	To determine the user's MH directory
Current-Folder:	To find the default current folder

Defaults

Context

If a folder is specified, it will become the current folder, and the message that is shown (i.e., the next message in sequence) will become the current message.

NAME

pick − select messages by content

SYNOPSIS

pick { −cc | −date | −from | −search | −subject | −to | −−component } [−src +folder] [msgs] [−help] [−scan] [−noscan] [−show] [−noshow] [−nofile] [−nokeep] pattern [−file [−preserve] [−link] +folder ... [−nopreserve] [−nolink]] [−keep [−stay] [−nostay] [+folder ...]]

typically:

```
pick −from jones −scan
pick −to holloway
pick −subject ned −scan −keep
```

DESCRIPTION

Pick searches messages within a folder for the specified contents, then performs several operations on the selected messages.

A modified *grep*(I) is used to perform the searching, so the full regular expression (see *ed*(I)) facility is available within 'pattern'. With '−search', pattern is used directly, and with the others, the grep pattern constructed is:

"component:.*pattern"

This means that the pattern specified for a '−search' will be found everywhere in the message, including the header and the body, while the other search requests are limited to the single specified component. The expression '−−component pattern' is a shorthand for specifying '−search "component:.*pattern" '; it is used to pick a component not in the set [cc date from subject to]. An example is "pick −−reply−to pooh −show".

Searching is performed on a per-line basis. Within the header of the message, each component is treated as one long line, but in the body, each line is separate. Lower-case letters in the search pattern will match either lower or upper case in the message, while upper case will match only upper case.

Once the search has been performed, the selected messages are scanned (see *scan*) if the '−scan' switch is given, and then they are shown (see *show*) if the '−show' switch is given. After these two operations, the file operations (if requested) are performed.

The '−file' switch operates exactly like the *file* command, with the same meaning for the '−preserve' and '−link' switches.

The '−keep' switch is similar to '−file', but it produces a folder that is a subfolder of the folder being searched and defines it as the current folder (unless the '−stay' flag is used). This subfolder contains the messages which matched the search criteria. All of the MH commands may be used with the sub-folder as the current folder. This gives the user considerable power in dealing with subsets of messages in a folder.

The messages in a folder produced by '−keep' will always have the same numbers as they have in the source folder (i.e., the '−preserve' switch is automatic). This way, the message

numbers are consistent with the folder from which the messages were selected. Messages are not removed from the source folder (i.e., the '−link' switch is assumed). If a '+folder' is not specified, the standard name "select" will be used. (This is the meaning of "(select)" when it appears in the output of the *folder* command.) If '+folder' arguments are given to '−keep', they will be used rather than "select" for the names of the subfolders. This allows for several subfolders to be maintained concurrently.

When a '−keep' is performed, the subfolder becomes the current folder. This can be overridden by use of the '−stay' switch.

Here's an example:

```
 1 % folder +inbox
 2       inbox+ has  16 messages (  3− 22); cur=  3.
 3 % pick −from dcrocker
 4 6 hits.
 5 [+inbox/select now current]
 6 % folder
 7  inbox/select+ has   6 messages (  3− 16); cur=  3.
 8 % scan
 9   3+  6/20  Dcrocker        Re: ned file update issue...
10   6   6/23  Dcrocker        removal of files from /tm...
11   8   6/27  Dcrocker        Problems with the new ned...
12  13   6/28  d crocker       newest nned ≪I would ap...
13  15   7/ 5  Dcrocker        nned ≪Last week I asked...
14  16   7/ 5  d crocker       message id format ≪I re...
15 % show all | print
16   [produce a full listing of this set of messages on the line printer.]
17 % folder −up
18        inbox+ has  16 messages (  3− 22); cur=  3; (select).
19 % folder −down
20  inbox/select+ has   6 messages (  3− 16); cur=  3.
21 % rmf
22 [+inbox now current]
23 % folder
24        inbox+ has  16 messages (  3− 22); cur=  3.
```

This is a rather lengthy example, but it shows the power of the MH package. In item 1, the current folder is set to inbox. In 3, all of the messages from dcrocker are found in inbox and linked into the folder "inbox/select". (Since no action switch is specified, '−keep' is assumed.) Items 6 and 7 show that this subfolder is now the current folder. Items 8 through 14 are a *scan* of the selected messages (note that they are all from dcrocker and are all in upper and lower case). Item 15 lists all of the messages to the high-speed printer. Item 17 directs *folder* to set the current folder to the parent of the selection-list folder, which is now current. Item 18 shows that this has been done. Item 19 resets the current folder to the selection list, and 21 removes the selection-list folder and resets the current folder to the parent folder, as shown in 22 and 23.

Files

$HOME/. mh_profile	The user profile

Profile Components

Path:	To determine the user's MH directory
Folder−Protect:	For protection on new folders
Current-Folder:	To find the default current folder

Defaults

'−src +folder' defaults to current
'msgs' defaults to all
'−keep +select' is the default if no '−scan', '−show', or '−file' is specified

Context

If a '−src +folder' is specified, it will become the current folder, unless a '−keep' with 0 or 1 folder arguments makes the selection-list subfolder the current folder. Each selection-list folder will have its current message set to the first of the messages linked into it unless the selection list already existed, in which case the current message won't be changed.

NAME

prev − show the previous message

SYNOPSIS

prev [+folder] [−switches for *l*] [−help]

DESCRIPTION

Prev performs a *show* on the previous message in the specified (or current) folder. Like *show*, it passes any switches on to the program *l*, which is called to list the message. This command is exactly equivalent to "show prev".

Files

$HOME/.mh_profile	The user profile

Profile Components

Path:	To determine the user's MH directory
Current-Folder:	To find the default current folder

Defaults

Context

If a folder is specified, it will become current, and the message that is shown (i.e., the previous message in sequence) will become the current message.

NAME

prompter − prompting editor front end

SYNOPSIS

This program is not called directly but takes the place of an editor and acts as an editor front end.

prompter [−erase chr] [−kill chr] [−help]

DESCRIPTION

Prompter is an editor which allows rapid composition of messages. It is particularly useful to network and low-speed (less than 2400 baud) users of MH. It is an MH program in that it can have its own profile entry with switches, but it can't be invoked directly as all other MH commands can; it is an editor in that it is invoked by an "−editor prompter" switch or by the profile entry "Editor: prompter", but functionally it is merely a text-collector and not a true editor.

Prompter expects to be called from *comp*, *repl*, *dist*, or *forw*, with a draft file as an argument. For example, "comp −editor prompter" will call *prompter* with the file "draft" already set up with blank components. For each blank component it finds in the draft, it prompts the user and accepts a response. A <RETURN> will cause the whole component to be left out. A "\" preceding a <RETURN> will continue the response on the next line, allowing for multiline components.

Any component that is non-blank will be copied and echoed to the terminal.

The start of the message body is prompted by a line of dashes. If the body is non-blank, the prompt is "--------Enter additional text". Message-body typing is terminated with a <CTRL-D> (or <OPEN>). Control is returned to the calling program, where the user is asked "What now?". See *comp* for the valid options.

The line editing characters for kill and erase may be specified by the user via the arguments "−kill chr" and "−erase chr", where chr may be a character; or "\nnn", where nnn is the octal value for the character. (Again, these may come from the default switches specified in the user's profile.)

A <DEL> during message-body typing is equivalent to <CTRL-D> for compatibility with NED. A <DEL> during component typing will abort the command that invoked *prompter*.

Files

None

Profile Components

prompter-next:	To name the editor to be used on exit from *prompter*

Defaults

Context

None

NAME

repl − reply to a message

SYNOPSIS

repl [+folder] [msg] [−editor editor] [−inplace] [−annotate] [−help] [−noinplace] [−noannotate]

DESCRIPTION

Repl aids a user in producing a reply to an existing message. In its simplest form (with no arguments), it will set up a message-form skeleton in reply to the current message in the current folder, invoke the editor, and send the composed message if so directed. The composed message is constructed as follows:

To: <Reply-To> or <From>
cc: <cc>, <To>
Subject: Re: <Subject>
In-reply-to: Your message of <Date>
<Message-Id>

where field names enclosed in angle brackets (< >) indicate the contents of the named field from the message to which the reply is being made. Once the skeleton is constructed, an editor is invoked (as in *comp*, *dist*, and *forw*). While in the editor, the message being replied to is available through a link named "@". In NED, this means the replied-to message may be "used" with "use @", or put in a window by "window @".

As in *comp*, *dist*, and *forw*, the user will be queried before the message is sent. If '−annotate' is specified, the replied-to message will be annotated with the single line

Replied: ≪Date≫.

The command "comp −use" may be used to pick up interrupted editing, as in *dist* and *forw*; the '−inplace' switch annotates the message in place, so that all folders with links to it will see the annotation.

Files

$HOME/. mh_profile	The user profile
<mh-dir>/draft	The constructed message file
/usr/bin/send	To send the composed message

Profile Components

Path:	To determine the user's MH directory
Editor:	To override the use of /bin/ned as the default editor
Current-Folder:	To find the default current folder

Defaults

'+folder' defaults to current
'msgs' defaults to cur
'−editor' defaults to /bin/ned
'−noannotate'
'−noinplace'

Context

If a '+folder' is specified, it will become the current folder, and the current message will be set to the replied-to message.

NAME

rmf − remove folder

SYNOPSIS

rmf [+folder] [−help]

DESCRIPTION

Rmf removes all of the files (messages) within the specified (or default) folder, and then removes the directory (folder). If there are any files within the folder which are not a part of MH, they will *not* be removed, and an error will be produced. If the folder is given explicitly or the current folder is a subfolder (i.e., a selection list from *pick*), it will be removed without confirmation. If no argument is specified and the current folder is not a selection-list folder, the user will be asked for confirmation.

Rmf irreversibly deletes messages that don't have other links, so use it with caution.

If the folder being removed is a subfolder, the parent folder will become the new current folder, and *rmf* will produce a message telling the user this has happened. This provides an easy mechanism for selecting a set of messages, operating on the list, then removing the list and returning to the current folder from which the list was extracted. (See the example under *pick*.)

The files that *rmf* will delete are cur, any file beginning with a comma, and files with purely numeric names. All others will produce error messages.

Rmf of a read-only folder will delete the "cur−" entry from the profile without affecting the folder itself.

Files

$HOME/.mh_profile	The user profile

Profile Components

Path:	To determine the user's MH directory
Current-Folder:	To find the default current folder

Defaults

'+folder' defaults to current, usually with confirmation

Context

Rmf will set the current folder to the parent folder if a subfolder is removed; or if the current folder is removed, it will make "inbox" current. Otherwise, it doesn't change the current folder or message.

NAME

rmm − remove messages

SYNOPSIS

rmm [+folder] [msgs] [−help]

DESCRIPTION

Rmm removes the specified messages by renaming the message files with preceding commas. (This is the Rand-UNIX backup file convention.)

The current message is not changed by *rmm*, so a *next* will advance to the next message in the folder as expected.

Files

$HOME/. mh_profile	The user profile

Profile Components

Path:	To determine the user's MH directory
Current-Folder:	To find the default current folder

Defaults

'+folder' defaults to current
'msgs' defaults to cur

Context

If a folder is given, it will become current.

NAME

scan − produce a one-line-per-message scan listing

SYNOPSIS

scan [+folder] [msgs] [−ff] [−header] [−help] [−noff] [−noheader]

DESCRIPTION

Scan produces a one-line-per-message listing of the specified messages. Each *scan* line contains the message number (name), the date, the "From" field, the "Subject" field, and, if room allows, some of the body of the message. For example:

```
 #     Date     From          Subject   [≪Body]
15+    7/ 5     Dcrocker      nned  ≪Last week I asked some of
16 −   7/ 5     dcrocker      message id format  ≪I recommend
18     7/ 6     Obrien        Re: Exit status from mkdir
19     7/ 7     Obrien        "scan" listing format in MH
```

The '+' on message 15 indicates that it is the current message. The '−' on message 16 indicates that it has been replied to, as indicated by a "Replied:" component produced by an '−annotate' switch to the *repl* command.

If there is sufficient room left on the *scan* line after the subject, the line will be filled with text from the body, preceded by ≪. *Scan* actually reads each of the specified messages and parses them to extract the desired fields. During parsing, appropriate error messages will be produced if there are format errors in any of the messages.

The '−header' switch produces a header line prior to the *scan* listing, and the '−ff' switch will cause a form feed to be output at the end of the *scan* listing. See Appendix D.

Files

$HOME/. mh_profile	The user profile

Profile Components

Path:	To determine the user's MH directory
Current-Folder:	To find the default current folder

Defaults

Defaults:
'+folder' defaults to current
'msgs' defaults to all
'−noff'
'−noheader'

Context

If a folder is given, it will become current. The current message is unaffected.

NAME

send − send a message

SYNOPSIS

send [file] [−draft] [−verbose] [−format] [−msgid] [−help] [−noverbose] [−noformat] [−nomsgid]

DESCRIPTION

Send will cause the specified file (default <mh-dir>/draft) to be delivered to each of the addresses in the "To:", "cc:", and "Bcc:" fields of the message. If '−verbose' is specified, *send;* will monitor the delivery of local and net mail. *Send* with no argument will query whether the draft is the intended file, whereas '−draft' will suppress this question. Once the message has been mailed (or queued) successfully, the file will be renamed with a leading comma, which allows it to be retreived until the next draft message is sent. If there are errors in the formatting of the message, *send;* will abort with a (hopefully) helpful error message.

If a "Bcc:" field is encountered, its addresses will be used for delivery, but the "Bcc:" field itself will be deleted from all copies of the outgoing message.

Prior to sending the message, the fields "From: user", and "Date: now" will be prepended to the message. If '−msgid' is specified, then a "Message-Id:" field will also be added to the message. If the message already contains a "From:" field, then a "Sender: user" field will be added instead. (An already existing "Sender:" field will be deleted from the message.)

If the user doesn't specify '−noformat', each of the entries in the "To:" and "cc:" fields will be replaced with "standard" format entries. This standard format is designed to be usable by all of the message handlers on the various systems around the ARPANET.

If an "Fcc: folder" is encountered, the message will be copied to the specified folder in the format in which it will appear to any receivers of the message. That is, it will have the prepended fields and field reformatting.

If a "Distribute-To:" field is encountered, the message is handled as a redistribution message (see *dist* for details), with "Distribution-Date: now" and "Distribution-From: user" added.

Files

$HOME/.mh_profile	The user profile

Profile Components

Path:	To determine the user's MH directory

Defaults

'file' defaults to draft
'−noverbose'
'−format'
'−nomsgid'

Context

Send has no effect on the current message or folder.

NAME

show − show (list) messages

SYNOPSIS

show [+folder] [msgs] [−pr] [−nopr] [−draft] [−help] [*l* or *pr* switches]

DESCRIPTION

Show lists each of the specified messages to the standard output (typically, the terminal). The messages are listed exactly as they are, with no reformatting. A program called *l* is invoked to do the listing, and any switches not recognized by *show* are passed along to *l*.

If no "msgs" are specified, the current message is used. If more than one message is specified, *l* will prompt for a <return> prior to listing each message.

l will list each message, a page at a time. When the end of page is reached, *l* will ring the bell and wait for a <RETURN> or <CTRL-D>. If a <return> is entered, *l* will clear the screen before listing the next page, whereas <CTRL-D> will not. The switches to *l* are '−p#' to indicate the page length in lines, and '−w#' to indicate the width of the page in characters.

If the standard output is not a terminal, no queries are made, and each file is listed with a one-line header and two lines of separation.

If '−pr' is specified, then *pr*(I) will be invoked rather than *l*, and the switches (other than '−draft') will be passed along. "Show −draft" will list the file <mh-dir>/draft if it exists.

Files

$HOME/. mh_profile	The user profile
/bin/l	Screen-at-a-time list program
/bin/pr	*pr*(I)

Profile Components

Path:	To determine the user's MH directory
Current-Folder:	To find the default current folder

Defaults

'+folder' defaults to current
'msgs' defaults to cur
'−nopr'

Context

If a folder is given, it will become the current message. The last message listed will become the current message.

Appendix A

COMMAND SUMMARY[3]

comp [−editor editor] [−form formfile] [file] [−use] [−nouse] [−help]

dist [+folder] [msg] [−form formfile] [−editor editor] [−annotate] [−noannotate] [−inplace] [−noinplace] [−help]

file [−src +folder] [msgs] [−link] [−preserve] +folder ... [−nolink] [−nopreserve] [−file file] [−nofile] [−help]

folder [+folder] [msg] [−all] [−fast] [−nofast] [−up] [−down] [−header] [−noheader] [−total] [−nototal] [−pack] [−nopack] [−help]

forw [+folder] [msgs] [−editor editor] [−form formfile] [−annotate] [−noannotate] [−inplace] [−noinplace] [−help]

inc [+folder] [−audit audit-file] [−help]

next [+folder] [−switches for *f*] [−help]

pick { −cc, −date, −from, −search, −subject, −to, −−component } [−src +folder] [msgs] [−help] [−scan] [−noscan] [−show] [−noshow] [−nofile] [−nokeep] pattern [−file [−preserve] [−link] +folder ... [−nopreserve] [−nolink]] [−keep [−stay] [−nostay] [+folder ...]]

prev [+folder] [−switches for *f*] [−help]

prompter [−erase chr] [−kill chr] [−help]

repl [+folder] [msg] [−editor editor] [−inplace] [−annotate] [−help] [−noinplace] [−noannotate]

rmf [+folder] [−help]

rmm [+folder] [msgs] [−help]

scan [+folder] [msgs] [−ff] [−header] [−help] [−noff] [−noheader]

send [file] [−draft] [−verbose] [−format] [−msgid] [−help] [−noverbose] [−noformat] [−nomsgid]

show [+folder] [msgs] [−pr] [−nopr] [−draft] [−help] [*l* or *pr* switches]

[3]All commands accept a −help switch.

Appendix B

MESSAGE FORMAT

This section paraphrases the format of ARPANET text messages given in Ref. 6.

ASSUMPTIONS

(1) Messages are expected to consist of lines of text. Graphics and binary data are not handled.

(2) No data compression is accepted. All text is clear ASCII 7-bit data.

LAYOUT

A general "memo" framework is used. A message consists of a block of information in a rigid format, followed by general text with no specified format. The rigidly formatted first part of a message is called the header, and the free-format portion is called the body. The header must always exist, but the body is optional.

THE HEADER

Each header item can be viewed as a single logical line of ASCII characters. If the text of a header item extends across several real lines, the continuation lines are indicated by leading spaces or tabs.

Each header item is called a component and is composed of a keyword or name, along with associated text. The keyword begins at the left margin, may contain spaces or tabs, may not exceed 63 characters, and is terminated by a colon (:). Certain components (as identified by their keywords) must follow rigidly defined formats in their text portions.

The text for most formatted components (e.g., "Date:" and "Message-Id:") is produced automatically. The only ones entered by the user are address fields such as "To:", "cc:", etc. ARPA addresses are assigned mailbox names and host computer specifications. The rough format is "mailbox at host", such as "Borden at Rand-Unix". Multiple addresses are separated by commas. A missing host is assumed to be the local host.

THE BODY

A blank line signals that all following text up to the end of the file is the body. (A blank line is defined as a pair of <end-of-line> characters with *no* characters in between.) No formatting is expected or enforced within the body.

Within MH, a line consisting of dashes is accepted as the header delimiter. This is a cosmetic feature applying only to locally composed mail.

Appendix C

MESSAGE NAME BNF

msgs	:=	msgspec | msgs msgspec
msgspec	:=	msg | msg-range | msg-sequence
msg	:=	msg-name | <number>
msg-name	:=	"first" | "last" | "cur" | "." | "next" | "prev"
msg-range	:=	msg"-"msg | "all"
msg-sequence	:=	msg":"signed-number
signed-number	:=	"+"<number> | "-"<number> | <number>

Where <number> is a decimal number in the range 1 to 999.

Msg-range specifies all of the messages in the given range and must not be empty.

Msg-sequence specifies up to <number> of messages, beginning with "msg" (in the case of first, cur, next, or <number>), or ending with "msg" (in the case of prev or last). +<number> forces "starting with msg", and -<number> forces "ending with number". In all cases, "msg" must exist.

Appendix D

EXAMPLE OF SHELL COMMANDS

UNIX commands may be mixed with MH commands to obtain additional functions. These may be prepared as files (known as shell command files or shell scripts). The following example is a useful function that illustrate the possibilities. Other functions, such as copying, deleting, renaming, etc., can be achieved in a similar fashion.

HARDCOPY

The command:

```
(scan -ff -header; show all -pr -f) | print
```

produces a scan listing of the current folder, followed by a form feed, followed by a formatted listing of all messages in the folder, one per page. Omitting "-pr -f" will cause the messages to be concatenated, separated by a one-line header and two blank lines.

You can create variations on this theme, using *pick*.

REFERENCES

1. Crocker, D. H., J. J. Vittal, K. T. Pogran, and D. A. Henderson, Jr., "Standard for the Format of ARPA Network Test Messages," *Arpanet Request for Comments*, No. 733, Network Information Center 41952, Augmentation Research Center, Stanford Research Institute, November 1977.

2. Thompson, K., and D. M. Ritchie, "The UNIX Time-sharing System," *Communications of the ACM*, Vol. 17, July 1974, pp. 365-375.

3. McCauley, E. J., and P. J. Drongowski, "KSOS—The Design of a Secure Operating System," *AFIPS Conference Proceedings*, National Computer Conference, Vol. 48, 1979, pp. 345-353.

4. Crocker, David H., *Framework and Functions of the "MS" Personal Message System*, The Rand Corporation, R-2134-ARPA, December 1977.

5. Thompson, K., and D. M. Ritchie, *UNIX Programmer's Manual*, 6th ed., Western Electric Company, May 1975 (available only to UNIX licensees).

6. Bilofsky, Walter, *The CRT Text Editor NED—Introduction and Reference Manual*, The Rand Corporation, R-2176-ARPA, December 1977.

RAND/R-2367-AF