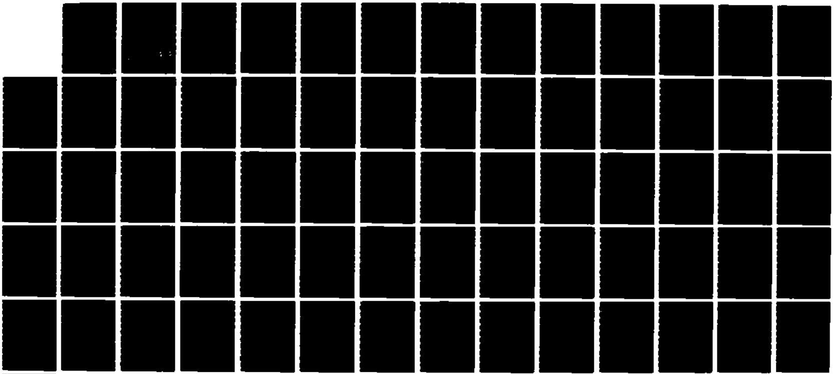
AD-A137 305 AN INTRODUCTORY GUIDE TO TOPS-20(U) UNIVERSITY OF SOUTHERN CALIFORNIA MARINA DEL REY INFORMATION SCIENCES INST L MOSES JUN 82 ISI/TM-82-22 MDA903-81-C-0335 1/1

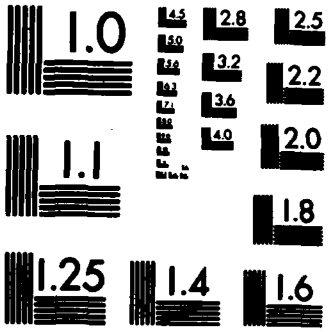
UNCLASSIFIED F/G 9/2 NL



END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

12

ISI/TM-82-22
June 1982

An Introductory Guide to TOPS-20

Lisa Moses

INFORMATION SCIENCES INSTITUTE
UNIVERSITY OF SOUTHERN CALIFORNIA
4676 Admiralty Way/Marina del Rey/California 90291
(213) 822-1511

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER ISI/TM-82-22	2. GOVT ACCESSION NO. AD-A137305	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) An Introductory Guide to TOPS-20		5. TYPE OF REPORT & PERIOD COVERED Technical Manual
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Lisa Moses		8. CONTRACT OR GRANT NUMBER(s) MDA903 81 C 0335
9. PERFORMING ORGANIZATION NAME AND ADDRESS USC/Information Sciences Institute 4676 Admiralty Way Marina del Rey, CA 90291		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Blvd. Arlington, VA 22209		12. REPORT DATE June 1982
		13. NUMBER OF PAGES 67
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) ----------		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This document is approved for public release and sale;
distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

----

18. SUPPLEMENTARY NOTES

----

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Emacs, Exec, Hermes, MM, Scribe, TOPS-20, user guide, XED

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This basic instructional manual introduces the use of several widely used programs on the DARPA sponsored TOPS-20 Computer Systems at USC/Information Sciences Institute. The programs covered are the Hermes message program, the MM message program, the XED text editor, the Emacs text editor, the Scribe formatting program, and some Exec level commands. This manual is intended for people who may not have experience with TOPS-20 or with programming.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ISI/TM-82-22
June 1982

An Introductory Guide to TOPS-20

Lisa Moses

INFORMATION SCIENCES INSTITUTE

UNIVERSITY OF SOUTHERN CALIFORNIA

4676 Admiralty Way/ Marina del Rey/ California 90291
(213) 822-1511

This research is supported by the Defense Advanced Research Projects Agency under Contract No. MDA903 81 C 0335. Views and conclusions contained in this report are the author's and should not be interpreted as representing the official opinion or policy of DARPA, the U.S Government, or any person or agency connected with them.

Messages regarding general questions, suggestions, and documentation should be addressed to ACTION@USC-ISI. In critical situations, users may call (213) 822-1511 and ask for ACTION.

If you have any problems with ACTION, please send a message to Jayne Millar, <MILLAR@USC-ISIB>, Manager User Services.

Contents

1. INTRODUCTION

This is an introductory document on the DARPA-sponsored TOPS-20 Systems at USC/Information Sciences Institute. It is intended for people who would like to use a computer for electronic mail, text editing, and document preparation, but who may not have experience with TOPS-20 or with programming. It does not include an in-depth explanation of the programs covered, but instead shows how to use the basic commands.

With the programs presented here, there is often more than one way to achieve the same results. I introduce the simplest set of commands to carry out a task, the assumption being that the user may proceed to more complex documentation after learning these basic commands.

In this document, the underlined words are what you type to execute commands. Underlined words that are also italicized are not meant to be typed literally, but represent individual names. For example, when you see

Log⇋*DirectoryName*⇋*Password*↵

you should type your own DirectoryName and Password, rather than the words "DirectoryName" and "Password." Another word that you will often see (underlined and italicized) is FileName. You must always substitute the actual name of the file that you are working with for the word FileName.

Generally, the System's response to your typed command is not shown, but only each sequence that you must type. Usually, I have not shown complete commands but rather the minimum that has to be typed before using the Escape Key.

The instructions for most programs are divided into "Lessons." Each lesson should be absorbed and understood before progressing to the next one. If this is your first experience with programs like these, you should review all of these lessons periodically and you'll find that you understand more with each review.

2. DEFINITIONS

Carriage Return

The Carriage Return is represented in this document by ↵. It confirms a given command and most commands are not executed until you type a carriage return. If you type a command and the cursor sits blinking after the last character typed, you probably need to type a carriage return.

Control Key

The Control Key is represented in this document by ↑. It is used with another key to enter a command. You may hold the Control Key down for as long as you want, but you should press the other key only once (the way you would use the Shift Key on a typewriter).

Cursor The Cursor is the small blinking line (about the size of a hyphen) that indicates your position on the terminal screen.

Default Whenever the System does something that is assumed, rather than something you've specified, that is the Default. For example, if you give the command to see the contents of a file (by typing Type FileName ↵), the System will assume that the file is in your Directory. That is the Default. If you want to see a file in another Directory, you must override the default by specifying the Directory before the FileName. When you want something other than the System Default, you must type more information.

Directory A Directory is a collection of files. Your Directory is your working space on the System. You can think of it as your office with your files in it. When you login (identify yourself to the System by giving your DirectoryName and Password), you are given access to your Directory and files. Whenever you see a name enclosed by angle brackets like this

```
<Name>
```

that name is a DirectoryName.

Escape Key
The Escape character is represented in this document by ⇋ . The Escape Key usually completes a command or a FileName. As soon as you have typed enough of a command or a FileName for the System to recognize it, type the Escape Key to complete it. If you haven't typed enough, the System simply beeps and waits for you to type more. At the Exec level it is not necessary to use the Escape Key on commands as long as you type a space instead. When you use the Escape key, though, you are prompted for the next part of the command. The System does require that DirectoryNames and FileNames be typed completely before it recognizes them, so the Escape Key saves you the effort of typing out the complete name yourself.

File A file is a specific collection of data such as a letter, a list, or a report. Files have names to enable the computer to recognize them. A file can be long (an entire document) or short (one line).

FileNames A FileName is the name of a file and is structured in a specific way. A FileName is composed of three parts: Name.Type.Generation . These parts are separated by periods. **No spaces are permitted in FileNames.** When you name a file, it is important to choose a name that is unique (so you don't have to type the whole name before typing the Escape Key) and that also reminds you of what is in the file. Quite often, a FileName is longer than one word. Since you may not use spaces to separate the words, you may use hyphens. You may also put one or more words in the Type. Sometimes, though, the FileName Type is significant and indicates an important aspect of the file. This is particularly true with Scribe (a formatting program) files. Generations are automatically given to files by the System. Each time a file is updated, it is given a higher number, so the most recent version of a file has the highest Generation number.

Password Your Password is the key to your Directory. When you login, you type your DirectoryName and then your Password. When you type your Password, it will not appear on the terminal screen. This way, you may keep your Password a secret.

Program Programs issue instructions to the computer. Many programs process files. For example a **message program** can, by following your commands, create and send a message and manipulate a file of messages. A **text editing program** can follow commands to create a text file and edit it, while a **text formatting program** can follow commands to format text by centering a heading, enumerating a list, italicizing a name, etc. Programs are stored as files on the System.

TOPS-20, Version 4 (Exec)

TOPS-20[1] is the main program or operating system of the computer. It interprets all of the commands and programs that you use. It is commonly referred to as the **System**. When you login, you are at the Exec level of TOPS-20. From the Exec level you enter different programs, and when you leave a program you return to the Exec level. You logout from the Exec level. When you are at the Exec level, you should always see an "@" sign on the left side of the screen. This is the Exec level "prompt" and simply indicates that you are at the Exec level.

[1] TOPS-20 is a product of Digital Equipment Corporation.

3. EXEC -- BASIC COMMANDS

Key:

↑	=	Control Key
⇋	=	Escape Key
↲	=	Carriage Return

3.1. EXEC - LESSON ONE

@ The @ is the prompt for the Exec level. Any time you type a carriage return and the System responds with @, you can be sure you are at the Exec level. In order to login or give any commands at the Exec level you must have the @ prompt. If you have not logged in, type ↑C to get the @ prompt.

↑C This is the first command you give before you login in order to get the @ prompt. Once you login, you may use this to abort a partially typed or partially completed command (sometimes you must type several ↑C's). If you are in another program and type ↑C, you will return to the Exec level (this is often not the best way to leave a program).

Login To login to the System, type Log⇋ *DirectoryName*⇋ *Password*↲ . Your password will not appear on the screen when you type it. You must login to the System before you can do any work.

Logout To logout, type Logo⇋↲ . When you are finished working for the day, you should logout of the System.

Directory To see the names of all the files in your Directory, type Dir⇋↲

↑T Use ↑T to see the current load average of the System (how busy the computer is). You may give this command from the Exec level and in most programs without interrupting the program. (Emacs is the only program covered in this document in which you may not type ↑T to see the load average.) When the load average is high (above 10), the System will respond more slowly to the commands you give. If there is no response at all when you type ↑T, the System may not be working (often referred to as a System "crash").

Type To see the contents of a file on your terminal screen:
Typ⇋ *FileName*⇋)

↑Q When the typeout of a file or message has stopped because the screen is full, type ↑Q to continue.

Del Key The Del Key deletes characters from right to left. You may need to use it with the Shift Key.

↑W ↑W deletes the word to the left of the cursor.

↑U ↑U deletes the entire line from the cursor to the left side of the screen.

? To see all of the commands available at the Exec level, type ?.

3.2. EXEC - LESSON TWO

Copy To copy a file from one Directory to your logged in Directory (on the same computer): Cop⇋ *<DirectoryName>FileName⇋⇋*↵

Rename To rename a file: Ren⇋ *OldFileName⇋ NewFileName*↵(no ⇋)

List/Print (Lineprinter)

To send a file to the Lineprinter: List⇋ *FileName⇋*↵ or Print⇋ *FileName⇋*↵

Xpress (Penguin)[2]

To send a file to the Penguin printer: Xpress⇋ *FileName⇋*↵

To cancel a job that you've sent: Cancel Xpress *#* ↵ (find out the Req # by checking the Print Queue).

To see other options you may use with the Xpress command, type Xpress⇋ *FileName⇋/?*

Print Queues When you give the List or Xpress commands, your job may not be printed immediately if other jobs are also waiting to be printed. All of the jobs are in a Queue until they are printed. To see the Lineprinter and Penguin Queues, type i⇋o⇋↵

Change Password

To change your password:
Set⇋ Dir⇋ Pas⇋ *DirectoryName*↵ *OldPassword*↵ *NewPassword*↵ *NewPassword*↵

Systat This command, followed by a DirectoryName, will tell you if a user is logged in, and, if so, the user's job number and the program he or she is using (for example, Systat Lisa↵) . To see this information for everyone currently on the System, give the command without a DirectoryName after it. If you give this command followed by a DirectoryName and the System responds with only an @, the user is not logged in.

[2] The Penguin is a printer used at ISI and ARPA.

Talk

To talk (link) to someone else also logged in on the computer, type Talk *DirectoryName*↵. Now everything that appears on your terminal will appear on the screen of the person you're linked to, and vice-versa. Be sure to type a ; (semicolon) at the beginning of each line of dialogue. To end the link, type (with no ;) Break⇋↵. **Use the Systat command before you link, and link only to someone who is at the Exec level.**

↑F

This works like the Escape Key on FileNames, but completes only one part of the FileName at a time.

↑V

To type out a FileName that starts with a nonalphabetic character (like [or ↑) type a ↑V first, then the FileName. The ↑V will not show on the screen. Some FileNames contain more than one nonalphabetic character but you may only need to type one ↑V if you use the Escape Key to complete the FileName.

3.3. EXEC - LESSON THREE

Detach To Detach from the System: Det⇋↵ When you login, the System assigns you a job number that is associated with the terminal you are using. When you detach, the terminal is released but the job number is still associated with you. When you want to work on the System again, you should **attach** to that job rather than login a second job. Generally, the detach command is useful for changing terminals or for leaving the System temporarily. Use Logout to leave the System overnight.

Attach To return to the System after detaching: Att⇋ *DirectoryName*↵ *Password*↵

Take If you have a file in your directory named Login.Cmd, then the System reads it automatically when you login. This file may contain information about such things as the size of your terminal screen so that the System knows how much of a large file or message to print out before pausing. The System does not read this file automatically when you attach. To make the system read your Login.Cmd file after you attach, type Take Login.Cmd⇋↵

Connect To connect to another Directory:
Conn⇋ *DirectoryName*↵ *Password*↵
To return to your own Directory just type Conn↵

Directory Subcommands

To get information about a file (for example, size, time of creation, protection status), type Di⇋ *FileName* ⇋,↵. You will then have a @@ prompt and you can type a ?↵ for a list of available commands. The next paragraph is an example of how to use a directory subcommand.

File Protection (Directory Subcommand)

Di⇋ *FileName* ⇋,↵
@@Prot↵↵
FileName;P777752 (This number indicates an unprotected file.)
FileName;777704 or 775200 (Either of these numbers indicates a protected file.)

Changing File Protection

Type from the Exec level:
Set File Prot⇋ *FileName*⇋ *777752*↲ (for unprotected file) or *777704*↲ or *775200*↲ (for protected file).

Continue To return to the last program you were in, type C↲ (this is not ↑C). This is particularly useful if you were in a text editor (XED or Emacs) and you left by mistake without saving what you were working on. This command won't work after you've logged out and then logged in again, or if the System has crashed and then started up again.

3.4. EXEC - LESSON FOUR

All of the following commands are given at the Exec level:

Delete Deletes specified generations of individual files. Type Delete *FileName* ⇋ ↵ to delete the lowest (oldest) generation. Deleted files may be retrieved (with the Undelete command) until you Logout or Expunge (or the System crashes).

Undelete Brings back specified deleted files unless you have logged out or used the Expunge command. Type Undelete *FileName* ⇋ ↵ .

Expunge Permanently removes all deleted files. Type Expunge↵ . Don't specify a *FileName* because expunge works on all deleted files. Logging out will also expunge any deleted files.

Delver.Exe Deletes all generations of specific files except the highest (most recent) generation and/or the second highest, and the lowest generation. Type Delver⇋↵ . Respond to the questions about which generations to delete with "Y" or "N" (no carriage returns). When asked for a FileName, you can either respond with a FileName and the program will delete generations of that file only, or type a carriage return without giving a FileName to have the program work on all your files.

Migrated Files Once a month, files in your Directory that have not been touched (looked at or edited) are "Migrated," or stored on tape off the computer. This happens on a Saturday, and the following Monday you will find a message listing the files that were Migrated on the previous Saturday. You will still see the FileNames (when you use the Directory command) with the word "OFFLINE" next to them. You may delete Migrated files without Retrieving them first.

Archived Files If you would like to store a file offline (on tape, off the computer), you may type Archive⇋ *FileName*⇋↵ . The FileName will no longer appear in your Directory. You may see the list of Archived files in your Directory by using the directory subcommand "Archive" (see Exec-Lesson Three).
Caution: Don't Archive Mail files that you may want to Retrieve and add to later.

Retrieving Files You can retrieve Archived and Migrated files by typing at the Exec level: Retrieve⇋ *FileName*⇋↵ . When your file has been retrieved, you will receive a message to that effect.

4. HERMES -- BASIC COMMANDS

Hermes is a program designed by Bolt Beranek and Newman Inc. (BBN) to create and manipulate messages. The commands included here are only a small sample of the commands available, but they will allow you to create, send, read, and file your mail on the System.

Figure 4-1 below shows a typical message. Each part of the message is called a "**Field**" (To Field, Subject Field, Date Field, etc.). The Fields shown in this particular message (ending with and including the To Field) are what make up the message "**Header**." Sometimes, you may wish to see only the header information in a message. When you give the command for this (Survey), the same information will be shown in a different format. These different formats for showing messages are called "**Templates**." Templates are also used for composing messages. Hermes starts you out with a set of standard templates. Later on, you may want to customize Hermes by creating your own templates.

```
Message 20; 205 chars
Mail from ARPANET host USC-ISIF rcvd at 28-Dec-81 1132-PST
Date: 28 Dec 1981 1125-PST
Sender: NDiamond
Subject: Budget Questions
From: Clive Davis
To: AFranklin

Can you see me tomorrow at 2:00 about my budget questions?

Thanks,
CD
```

Figure 4-1: Sample message

The existing BBN documentation on Hermes is quite comprehensive and should be referred to once the material here has been covered [Mooers 82a, Mooers 82b].

4.1. HERMES - LESSON ONE

Key:

↑	=	Control Key
⇆	=	Escape Key
↵	=	Carriage Return

Hermes Enter Hermes from the Exec level by typing Hermes↵.

> The > is the Hermes prompt. When you see the > on the left side of the screen, you are in Hermes.

Mail.Txt The file Mail.Txt contains your current messages. It is created by the System when you receive your first message. When you enter Hermes, this file is automatically loaded into Hermes for you to read or modify.

Survey To see your message headers (message number, date, sender, and subject) in the order that the messages arrived, type Survey↵. To see your headers in reverse order (most recent messages first), type Survey Reverse↵. To see other commands that can be used with Survey, type Survey ?↵, and Hermes will print a list.

Print To read a message, type Print followed by the message number (for example, Print 5↵ to see message 5). You may read your next message by typing the Line Feed Key (or ↑J), and read your previous message by typing the Up Arrow Key (∧).

Quit To leave Hermes and return to Exec, type Quit↵.

? Type ? to see the commands that are available at any point in Hermes and a list will be printed on your screen. Once the list is printed, you will be left where you were when you typed the ?.

Describe To find out more in-depth information about a command or topic, type Describe and the command or topic. For example, type Describe Survey↵ for information on the Survey Command.

↑O Use ↑O to stop Hermes from printing a message or message headers.

↑U Use ↑U to abort a command.

↑E Use ↑E for panic aborts, when you realize Hermes is doing something you can't otherwise figure out how to stop.

General Information

Two Exec level commands also work in Hermes:

↑Q When the typeout of a message has stopped because the screen is full, type ↑Q to continue.

↑T To see the current load average of the System.

For all commands that require a message number, you may also give several numbers separated by commas (for example, print 1,3,7) or, for consecutive messages, the first and last numbers separated by a colon (for example, survey 5:7).

CAUTION: Never edit a mail file with a text editor. Always use Hermes commands to delete or edit messages.

4.1.1. The Compose Command

The Compose Command is used to create a draft message to send to other people.

To use the Compose Command type the UNDERLINED words as follows:

>Compose ↵

To: *DirectoryName* ↵ To send a message to more than one person, separate each DirectoryName with a comma. Don't use <> around the DirectoryNames.

Cc: *DirectoryName* ↵ This is for carbon copies. Put your DirectoryName here if you want a carbon copy of the message.

Subject: Oh What a Message! ↵

Text:
Whatever you type here will be the body of your message. You can make this as long or as short as you like.

To end your message, type ↑Z.

Format To format your text, type ↵↵ (two carriage returns).
If you don't want to format your text, type N↵ .

Send To send your message, type ↵↵ (two carriage returns).
If you don't want to send your message, type N↵ .

4.1.2. Modifying a Draft Message

If you say "no" to the send prompt, you will get the >> prompt. You can then use any of the following commands (to see other commands type a ?) :

Abort Typing Abort will abort the draft message and leave you at the > prompt. If you decide to Abort before you get to the >> prompt, type ↑E↵↵ and when you get the >> prompt, type Abort↵↵ .

From Adding a From: *Name* field indicates that you are sending a message for someone else. The name does not have to be a DirectoryName. Your name will still appear in the message header as "Sender."

Send Typing Send↲↲ queues the draft message to be sent when the Mailer program runs. If you want to send your draft message immediately, type Send ↲↲ and then when you have the > prompt, type Mailer↲. However, Mailer works automatically and need not be typed unless you are in a hurry.

Show Type Show to print the entire draft message on the screen.

Edit To edit your draft message with Emacs or XED, type Emacs↲ or XED↲ and you will enter the editor with the body of your draft message in the text buffer.[3] To return to Hermes from Emacs, type ↑X↑Z, and to return to Hermes from XED, type E↲.

Storing and Retrieving Draft Messages before Sending

If you have created a draft message but you do not want to send it immediately, you may store it as a file in your Directory and then retrieve and send it at a later time. To store your draft message before sending it, type (at the double prompt)

>> Store-Draft ⇋ *FileName* ↲
>> Done ↲.

To retrieve your stored message, type

> Restore-Draft⇋ *FileName* ↲.

You will then be at the >> prompt as if you had just created your draft message, and you may edit it or just type

>> Send↲↲.

4.1.3. Commands for Corrections

Del Key To delete one character at a time moving from right to left (may need the Shift Key).
↑W To delete one word at a time moving from right to left.
↑R To see a line with corrections, cleaned up.
↑U To delete the entire line from the cursor to the left side of the screen.

[3] Starting with Version 4.3.3, Hermes allows you to place the whole draft message in an editor and move from field to field, only returning to Hermes to send it.

4.2. HERMES - LESSON TWO

Key:

↑	=	Control Key
⇋	=	Escape Key
#	=	Message Number
↵	=	Carriage Return

Delete To delete a message, type Del #↵ . To delete a sequence of messages, type Del and the numbers of the first and last messages in the sequence separated by a : (Del 10:15↵ deletes messages 10 through 15). To delete messages not in sequence, separate each message number with a comma. Deleted messages can still be retrieved with the Undelete command as long as they have not been Expunged.

Undelete To undelete a deleted message, type Und #↵. You may also undelete more than one message at a time.

Expunge To **permanently** remove all deleted messages, type Expunge↵ . Your remaining messages will be renumbered and you will be left in Hermes.

List To get a hard copy of a message, type List #↵ . The default is for a Lineprinter listing of the message. For a Penguin listing, type List # ⇋⇋ PNG:↵ .

File This command is useful for collecting related messages into one file. To move messages from your Mail.Txt into another message file, type File # *FileName*⇋↵ . If this is a new file, type File # *FileName*↵↵ . Use the Get command described below.

Get To read a mail file other than Mail.Txt into Hermes, type Get *FileName*⇋↵ (for example, to read BBOARD, type Get <BBOARD>Mail.Txt⇋↵). To return to your Mail.Txt, type Get↵ .

4.3. HERMES - LESSON THREE

Key:

↑	=	Control Key
⇋	=	Escape Key
#	=	Message Number
↵	=	Carriage Return

Reply

To reply to a message you have received, the command is Reply #↵ . The "To," "From," and "Subject" fields will be completed by Hermes, and you will be prompted for "Text." Type your reply and end the message with ↑Z.

Forward

To send a copy of a message you have received to someone else, type Forward #↵ . You will be prompted for "To," "Subject," and "Text." The text you add will appear after the header but before the forwarded message.

Redistribute

This command allows you to send a copy of a message you have received to someone else, but without adding anything to the message (as in Forward). You will be prompted for "To" and "Refile." Answer yes to Refile, and your original message header will be changed to include "Redistributed-By" although the message number will remain the same in your Mail.txt. If you answer no to Refile, your original message will not indicate that it was redistributed, although the message that is sent will have the "Redistributed-By" field.

Switches

Switches determine how Hermes will work. You may set your own switches to the options you prefer. To see your switches as they are presently set, type Show Sw⇋↵ . To change the setting of your switches, find the switch you want changed using the Show Switches command, and type

Edit Switches⇋↵
Set *Switchname*⇋ to the value *option*↵
Done↵

5. MM -- BASIC COMMANDS

MM[1] is a program used to create and manipulate messages. The commands included here are only a small sample of the commands available, but they will allow you to create, send, read, and file your mail on the System.

The figure below shows a typical message. Each part of the message is called a **"Field"** (To Field, Subject Field, Date Field, etc.). The Fields shown in this particular message (ending with and including the To Field) are what make up the message **"Header."** Sometimes, you may wish to use the Headers command to see only the header information in a message.

```
#30 (191)
Date: 23 Mar 1982 1025-PST
From: Lisa Moses <LISA at USC-ISIB>
Subject: Budget Questions
To: QJones
Cc: CDavis

Can you see me tomorrow at 2:00 about my budget questions?

Thanks,
Lisa
-------
```

Figure 5-1: Sample message

More complete MM documentation is available on ISI machines in <Documentation>MM.Changes, MM.Doc, and MM.Info.

[4] MM was written by Michael McMahon, with contributions from several other individuals. MM is currently maintained by McMahon and Mark Crispin at Stanford University (Admin.MRC@SU-SCORE).

5.1. MM - LESSON ONE

Key:
↑ = Control Key
⇆ = Escape Key
↵ = Carriage Return

MM Enter MM from the Exec level by typing MM↵ .

MM> MM> is the MM prompt. Whenever you see MM> on the left side of the screen, you are in MM.

Mail.Txt The file Mail.Txt contains your current messages. It is created by the System when you receive your first message. When you enter MM, this file is automatically loaded into MM for you to read or modify.

Headers All To see your message headers (message number, date, sender, and subject) in the order that the messages arrived, type H A↵ . To see your headers in reverse order (most recent messages first), type H I↵ . To see other commands that can be used with Headers, type Headers ?↵ , and MM will print a list.

Read To read a message, type R followed by the message number (for example, R 5↵ to see message 5). You will now be in Read Mode and the prompt on the left margin will be R>. To see the commands available in Read Mode, type ? at the R> prompt. To leave Read Mode, type ↵ .

Quit To leave MM and return to Exec, type Q↵ .

↑N To abort the typeout of your Headers or of a message, type ↑N.

MM.Init This is a file that allows you to customize some aspects of MM. Create this file by entering MM and typing Create⇆↵ . Exit MM, and use a text editor to edit MM.init. For information on what your options are and how to set each option, read <Documentation>MM.Info.

? Type ? to see the commands that are available at any point in MM, and a list will be printed on your screen. Once the list is printed, you will be left where you were when you typed the ?.

Help To read about any of the commands listed when you type ?, type Help *Command*) . For example, type MM>Help Read) for information on the Read command.

General Information

Two Exec level commands also work in MM:

↑Q When the typeout of a message has stopped because the screen is full, type ↑Q to continue.

↑T To see the current load average of the System.

For all commands that require a message number, you may also give several numbers separated by commas (for example, read 1,3,7) or, for consecutive messages, the first and last numbers separated by a colon (for example, delete 5:7).

CAUTION: Never edit a mail file with a text editor. Always use MM commands to delete or edit messages.

5.1.1. Sending a Message

To use the Send Command, type the UNDERLINED words as follows:

MM>S↵

To: *DirectoryName* ↵ To send a message to more than one person, separate each DirectoryName with a comma. Don't use <> around the DirectoryNames.

Cc: *DirectoryName* ↵ This is for carbon copies. Put your DirectoryName here if you want a carbon copy of the message.

Subject: Oh What a Message! ↵

Text:
Whatever you type here will be the body of your message.

To end your message, type ↑Z.

5.1.2. Send Mode Commands

Once you type ↑Z to end your message, the prompt on the left side of the screen will be S> because you are in Send Mode. Now you may use any of the following commands (type a ? to see additional commands that can be used here):

Send To send your message, type ↵. **You may edit your MM.init file so that you must type Send↵ to send your message.**

Quit To abort your message, type Q↵.

Display To see your entire message before sending it, type Display↵.

Edit To edit your message in Emacs, type Edit↵ and you will enter Emacs with the body of your message in the main buffer. You may edit your message and then return to MM by typing ↑X↑Z.

From Adding a From: *Name* field indicates that you are sending a message for someone else. The name does not have to be a DirectoryName. Your name will still appear in the message header as "Sender." Do not type the colon when you are adding this field.

Storing and Retrieving Messages before Sending

If you have created a message but you do not want to send it immediately, you may store it as a file in your Directory and then retrieve and send it at a later time. To store your message before sending it, type (at the S> prompt)

```
S> Save-Draft ⇆ FileName↵
S> Q↵
```

To retrieve your stored message, type

```
MM> Restore-Draft⇆ FileName⇆↵
```

and you will be in the text field of your message. You may add to your message or type ↑Z to end it, and then edit or send it.

Commands for Corrections

Del Key	To delete one character at a time moving from right to left (may need the Shift Key).
↑W	To delete one word at a time moving from right to left.
↑U	To delete the entire line from the cursor to the left side of the screen.

5.2. MM - LESSON TWO

Key:

↑	=	Control Key
⇋	=	Escape Key
#	=	Message Number
↵	=	Carriage Return

Delete To delete a message, type Del #↵ . To delete a sequence of messages, type Del and the numbers of the first and last messages in the sequence separated by a : (Del 10:15↵ deletes messages 10 through 15). To delete messages not in sequence, separate each message number with a comma. Deleted messages can still be retrieved with the Undelete command as long as they have not been Expunged.

Undelete To undelete a deleted message, type Und #↵. You may also undelete more than one message at a time.

Expunge To **permanently** remove all deleted messages, type Expunge↵ . Your remaining messages will be renumbered and you will be left in MM.

List To get a Lineprinter listing of a message, type List #↵ .

5.3. MM - LESSON THREE

Key:
↑ = Control Key
⇋ = Escape Key
# = Message Number
↵ = Carriage Return

Reply

To reply to a message you have received, the command is Reply #↵ . MM will prompt you, "Send reply for message # to:" and you may type /s↵ to reply to the sender only, or type /a↵ to send a copy to everyone listed in the original message. (Type ? to see other options.) The "To," "From," and "Subject" fields will be completed by MM, and you will be prompted for "Text." Type your reply and end the message with ↑Z. Then type S> S↵ to send it.

Forward

To send a copy of a message you have received to someone else, type Forward #↵ . You will be prompted for "To" and "Text." If you want to add a message, type it here and end with ↑Z. To see the entire message before sending it, type (after ↑Z) S> Display↵ . Then type S> S↵ to send it.

Remail

This command allows you to send a copy of a message you have received to someone else without adding anything to the message (as in Forward). Type Remail #↵ . You will be prompted for "To" and your message will be sent immediately.

Move

This command is useful for collecting related messages into one file. To move messages from your Mail.Txt into another message file, type Move⇋*FileName*⇋ #↵ . If this is a new file, type Move⇋*FileName* #↵↵ . Use the Get command to see other message files.

Get

To read a mail file other than Mail.Txt into MM, type Get *FileName*⇋↵ . To return to your Mail.Txt, type Get↵ . To read BBOARD, you need only type BB↵.

6. XED -- BASIC COMMANDS

XED is a text editor. With XED you may create a text file (a letter, list, chart, report, etc.) and edit it at the time you create it, or store it and then edit it at a later time. XED is a **line editor**, and when you work in XED all of your lines are numbered so that it's easy to keep track of your position in the file. The line numbers are not actually part of the file and do not appear when the file is printed on paper.

After entering XED you are either in the "Command" mode, where every character you type is an XED command, or in "Text Entry" (Append) mode, where every character you type is entered into your file as text. Once you have mastered those two modes, you will then learn "Change" mode, which allows you to return to an already typed text line to edit it.

You may want to refer to some other documentation: the *XED User's Manual*, ISI/TM-76-3, and *the JOY of TENEX and TOPS-20*, volume 2, ISI/TM-79-16, by Chloe S. Holg.

6.1. XED - LESSON ONE

Key:

↑	=	Control Key
⇆	=	Escape Key
↵	=	Carriage Return

XED To enter XED type XED ↵. When you see a : (a colon, the XED command prompt) you may give your first command.

Append To type in text and thereby create a file, type A. When you see a * (an asterisk, the XED text prompt), you may begin to type in text. Lines are numbered automatically. If you type a carriage return without typing text on the line, you will create a blank line (for example, two carriage returns will create two blank lines).

↑Z Type ↑Z to return to the command level from Append.

Del Key and ↑W To correct mistakes as you type (in Append) use the Del Key to delete one character or space and ↑W to delete one word. Both the Del Key and ↑W work from right to left.

/ To see your current line (where you are in the file), type a / (forward slash). Another way to determine your current line is to look for the cursor. It should be immediately below your current line. To make another line your current line, type the number of the line you want to be current and then type a / (forward slash). This line becomes your current line. Most commands in XED work on the current line unless another line number is given immediately before the command.

View To see a large section of a file, type a V. This will print your current line and the next 20 lines. To see a file from the beginning, type 1V.

Exit To leave XED and name the file that you have created, type an E (from the command level), then type the FileName followed by ↵ (do not type ⇆). The file will now be in your Directory. To Exit an existing file (one that already has a name), type E⇆↵.

Read

To enter XED and see or edit a file that you have already created, type XED↲ and when you get the : prompt, type R *FileName* ⇋↲. When you leave XED this time, type E⇋↲. You may type an ⇋ instead of typing the FileName because XED knows the FileName. The System will automatically give the file a new generation number. The Read command can also be used to combine files by reading in one file and then reading in another. The second file will be inserted into the first, immediately below the current line, and you'll be left at the last line of the second file.

↑Q

To abort commands, type ↑Q. This will leave you at the command level. **Use ↑Q to get out of sticky situations.**

Kill

To eliminate one line, type the line number followed by K↲. To kill more than one line, type the number of the first line you want to kill, K: and the number of the last line you want to kill (for example, 15K:30↲).

Practice for Lesson One

Practice what you have learned by entering XED and using the Append command to create a file. Then Exit and name the file. Re-enter XED (from Exec), Read in that file, View it, and Exit from XED. Send the file to the Lineprinter from the Exec level by using the Exec command List *FileName*⇋ ↲.

6.2. XED - LESSON TWO

Key:
↑ = Control Key
⇋ = Escape Key
↵ = Carriage Return

Change To modify the current line, type C at the command level, and then use the subcommands below to delete and insert characters. To leave Change and return to the Command level, type a carriage return.

These subcommands are used within the Change command:

Skip To move to a specific character, type S, followed by the character.

Delete Once you have moved to a character you wish to delete, type D to delete it. This will delete the character directly over the cursor.

Insert To insert text, type i. Everything typed after the i will be inserted on the line. To end insert, type a carriage return (the terminal will beep). A second carriage return ends Change.

End To move to the end of a line, type E. You can then delete from right to left using the Del Key or the Backspace Key, ↑W (to delete one word), and ↑X (to delete everything to the left of the cursor). You may also insert text at this point.

Break Wherever this command is typed, the line you are working on will be broken into two and you will be left in Change at the beginning of the second line.

↑D Use ↑D to abort any changes you have made. You must use it while you are still on the line.

? To see all of the commands that are available in Change, type ?.

To leave Change and return to the command level, type one carriage return. If you are in Insert, type two carriage returns (the System will beep after the first one).

Practice for Lesson Two

Practice the commands from Lesson Two by entering XED and copying the following paragraph verbatim; then correct the mistakes on each line.

```
In ordder to makeany changes(either addingg ro deleting text))
tto a single line in XEDXED, you must Use thee Chaneg command.  Onec
  youhave tpyyed the Change command, the you mmay use the subcommands
listed on pgae 1.  There are usually severalways to delete characterrrs
on a line within the Change command adn You will gradually learnn
tochoose the the fastest commands forr each situation.
```

Once you have corrected the paragraph above, Exit from XED, name the file, and send it to the Lineprinter.

6.3. XED - LESSON THREE

Key:

↑	=	Control Key
⇋	=	Escape Key
↵	=	Carriage Return

Kill The Kill command is used to remove unwanted lines and also, along with the Jam command, to move text around in a file. To eliminate one line, type the line number followed by K↵. To kill more than one line, type the number of the first line you want to kill, K: and the number of the last line you want to kill (for example, 15K:30↵).

Print Dump Once you have killed a line (or a group of lines), it is stored in the Print Dump. To see what's in the Print Dump, type P at the command level. Only the most recently killed line (or group of lines) is stored in the Print Dump, so as soon as you use the kill command again, the newly killed text will replace what was in the Print Dump.

Text Buffer This is the working area in XED. When you are editing a file, it is in the Text Buffer. When you enter XED and start a new file with the Append command, you are also working in the Text Buffer.

Jam To put text from the Print Dump back into the Text Buffer, type a line number followed by J. This will Jam a copy of the text from the print dump *below* the line number given. You may Jam the same text in the print dump more than once. If you just type J without giving a line number, the text in the Print Dump will be inserted in your file below your current line. Use the Kill and Jam commands to move text around in a file.

Zap To put your entire file into the Print Dump, type Z. This allows you to clear your Text Buffer for another file without returning to the Exec level.

Switch Dump To switch the entire Text Buffer with the Print Dump, type ' (an apostrophe).

Practice for Lesson Three

Practice the commands from Lesson Three by entering XED and copying the following paragraph verbatim:

Kill and Jam commands. Remember that as soon as you kill a
To change the order of the lines in this paragraph, use the
To put the text in the Print Dump back into your file, use
line it is in the Print Dump until you kill another line.
the Jam command. This will insert the text below the current line.

Once you have copied the paragraph, use the Kill and Jam commands to put the lines in the following order:

To change the order of the lines in this paragraph, use the
Kill and Jam commands. Remember that as soon as you kill a
line it is in the Print Dump until you kill another line.
To put the text in the Print Dump back into your file, use
the Jam command. This will insert the text below the current line.

Once you have the lines in the correct order, Exit from XED, name the file, and send it to the Lineprinter.

6.4. OTHER USEFUL XED COMMANDS

The following commands supplement the commands presented in the three previous lessons.

Command Level Commands

I - Insert Insert is similar to the Append command, except that it allows you to input text starting with the line **before** the current line instead of the line after the current line. (Do not confuse this command with the Insert command in Change mode.) Return to the Command mode by typing ↑Z.

L - List List prints out the entire file on the terminal screen starting with line 1. It does not show line numbers.

Line Feed Key Typing the Line Feed Key prints the next line (↑J also works).

∧ (Up Arrow Key)

Typing the Up Arrow Key prints the previous line.

? Typing a Question Mark prints a list of all of the available commands on the screen.

B - Back-up Typing a B creates a back-up file. If the System crashes while you are working on a file in XED, you will lose whatever you have typed since the last time you created a new generation of that file. To prevent a great loss, use the Back-up command every page or so. If the System does crash, you will only lose what you have done since the last Back-up command was given.

$ - Dollar Sign Typing a dollar sign will move you to the last line of the file.

F - Find Typing F followed by a string of text (a letter, word, or phrase) and a carriage return, finds the next occurrence of that string, starting from wherever you are in the file.

S - Search This command works like Find except that Search finds all occurrences of specified text rather than one at a time.

X - Exchange This command finds each successive occurrence of a string of text and allows you to decide whether or not to exchange it for another string of text. Type X and it will ask "old" (give string to be changed) and then "new" (give desired replacement string). As it finds each occurrence, you may confirm that exchange by typing a carriage return; skip it by typing N ; or you may confirm all exchanges by typing * . To abort the exchange command, type ↑Q . This command may be used instead of the Change command to edit a line of text.

% - Sndmsg Typing % (the percent sign) invokes Sndmsg, a simple program that allows you to send a copy of the entire file in the Text Buffer as a message. You are prompted for the Subject, To, and Cc fields of the message.

, - (Comma) Typing a comma shows your current line in context by printing five lines before the current line, highlighting the current line, and printing five lines after the current line.

G - Group The Group command joins the current line with the following line.

" - Mode File The Mode File establishes how XED will work for you. A Mode File is created the first time you go into XED; you may also copy a Mode File from someone else. You can change this file to establish the size of the right margin, modify commands, or even turn off some commands completely. Change your Mode File by typing " (the quotation mark) and giving the option you prefer for the commands listed. When you see the question "Do you want to make these mode changes permanent?" type the Escape Key and a new generation of your XED Mode File will be created. If you decide you don't like the changes you made in your Mode File, you may either type the quotation mark again and make the changes you want, or just delete the highest generation of the Mode File at the Exec level. The name of the Mode File is $XED-MODE-FILE$.

@ - Lower Exec Typing the @ sign enters Exec on a lower fork. To return to XED, type Pop and you will return to the current line.

Q - Quit The Quit command leaves XED without creating a new generation of the file. If you have edited the file and typed Q or ↑C by mistake, and you suddenly find yourself at the Exec level, just type a C (for Continue) and you will be back in XED at the current line (whew). Quit is a good command to use if you enter an existing file and make a major editing mistake: use the Quit command, and the generation you are working on will **not** be saved. Then you can start over by entering XED and reading in the generation of the file you had before making the error.

7. EMACS -- BASIC COMMANDS

Emacs[5] is a very powerful text editor. It is a **screen editor**, so the lines are not numbered. All commands start with either the Escape Key or the Control Key. When you use the Escape Key, type it once, release it, then type the next character. When you use the Control Key, hold it down while you type the other character. Anything you type without the Escape Key or the Control Key will be entered as text. The cursor shows where text will be entered if you type, and the cursor can be moved anywhere on the screen.

Near the bottom of the screen is a line called the "Mode Line" containing information that is continuously updated. It may show the time and load average, the name of the file you are currently working on, where you are in the file at that moment, and whether the file has been changed since you last updated and saved it.

The area of the screen below the mode line often shows where you are in a command. For example, whenever you type the Escape Key, the characters "M-" appear on that line.[6] When you see "M-" you know that Emacs is going to consider the next character you type to be a command. It is important to pay attention to this part of the screen.

Emacs has many commands, and they can often be used in several ways to accomplish the same thing. The commands presented here will allow the user to do all the basic things that a text editor is needed for. After studying these commands, you can use a tutorial program called "Teach-Emacs" to review these commands and learn others. Type Teach-Emacs⇆↵ from the Exec level and follow the instructions. The Emacs manual is available in <Emacs>Emacs.guide on all ISI Systems.

[5] Emacs was written by Richard M. Stallman at the MIT Artificial Intelligence Lab, and is distributed on a basis of communal sharing. AI Lab memo 519a is available for people who want more information on how EMACS works.

[6] Emacs was written for terminals having a Meta Key. It was altered to work on terminals without a Meta Key so that the Escape Key works instead. Whenever you see "M-" on your screen or in Emacs documentation, it may indicate the Escape Key if your terminal does not have a Meta Key.

7.1. EMACS - LESSON ONE

Key:

↑	=	Control Key
⇋	=	Escape Key
↵	=	Carriage Return

Enter Emacs Edit↵

Create New File To create a new file, enter Emacs and start typing.

Name New File ↑X↑W *NewFileName*↵ This commands also writes out (saves) the file.

Update/Save File

↑X↑S This command creates a new generation of the file you are working on. You may also name a file with this command.

Read In Existing File

↑X↑F *FileName*⇋↵

Leave Emacs ↑X↑Z Returns you to the Exec level. Be sure that you have saved the current generation of the file (if you want it) before exiting.

Killing Lines and Regions

To kill one line, go to the beginning of the line and type ↑K. To kill a region (anything longer than a line), first mark the beginning of the region to be killed by moving the cursor to the first character of the region and typing ↑@ (the @ won't appear on the screen). Then move the cursor one character past the end of the region to be killed and type ↑W. To retrieve what has been killed, type ↑Y.

Abort Command Type ↑G to abort a command that has not yet been completed. Sometimes, you must type several ↑G's to abort.

Undo Last Command

Type ⇋X undo↵ to actually undo a command immediately after it is completed. If you mistakenly give a command that makes a great change in the text buffer, you can often undo the change without having to know precisely how it came about by using this command.

Repeating Commands

Most commands can be given so that they repeat a specified number of times. To repeat a command, type ⇋, then type the number of times you wish to repeat the command, then type the command. For example, to move forward 1 word in a file type ⇋F and to move forward 10 words type ⇋10 ⇋F .

Minibuffer

If you should happen to type ⇋ twice accidently, or if you hold the Escape Key down too long, you will enter the "Minibuffer." To get out of the Minibuffer, type ⇋ twice again, and you will be back where you were in your file.

DELETING TEXT

↑D	Delete next character
⇋D	Delete next word
Del	Delete previous character (may need the Shift Key)
⇋Del	Delete previous word (may need the Shift Key)

MOVING THE CURSOR

↑F	Move forward one character
↑B	Move back one character
↑N	Move to next line
↑P	Move to previous line
↑A	Move to beginning of line
↑E	Move to end of line
⇋A	Move to beginning of sentence
⇋E	Move to end of sentence
⇋<	Move to beginning of file
⇋>	Move to end of file
⇋F	Move forward one word
⇋B	Move back one word
↑V	Display next screen of text
⇋V	Display previous screen of text

7.2. EMACS - LESSON TWO

Key:
↑ = Control Key
⇋ = Escape Key
↲ = Carriage Return

Help

To read the help options, type ↑Del Key ? and follow the instructions to get information about command characters, functions, Emacs news, etc.

Searching For Text (Incremental Search)

To search forward in your file, type ↑S and then type the text you're searching for. End or abort the search by typing ⇋ . To find successive occurrences, repeat ↑S . To search backward in your file, use ↑R.

Exchanging Text (Query Replace)

This command finds each successive occurrence of a string of text and allows you to decide whether or not to exchange it for another string of text. Type ⇋% *existing text* (give current string to be changed) ⇋ *new text* (give desired string) ⇋⇋ to start the exchange. As each occurrence of existing text is found, you may have it replaced with new text by typing one space bar or skip it by typing Del Key (you may need the Shift Key). You may confirm all exchanges by typing ! . To abort the command, type ↑G .

Joining Files

To combine files, read one file into a buffer with ↑X↑F , move the cursor to the point where you want the second file to be, and then read the second file into the same buffer with ⇋X Insert File *FileName*↲ . The second file will be inserted immediately below the cursor.

Multiple Buffers

A buffer is a working space in Emacs. While you are editing a file, it is in a text buffer. Each buffer holds one file and has a separate name (usually the first part of the FileName). Emacs allows you to have multiple buffers and to switch back and forth between them. To read a file into a buffer, type ↑X↑F *FileName*⇋↲ . To see the names of the buffers and the names of the files in them, type ↑X↑B (notice the buffer names). Type ↲ to get back to your file. To switch from one buffer to another, type ↑XB *BufferName*↲ .

Split Screen To display two files simultaneously, read in one file with ↑X↑F *FileName*⇆↲ , create a second window with ↑X2 , and read in a second file with ↑X↑F *FileName*⇆↲ . To switch back and forth between windows, type ↑XO (the letter "O," not zero), and to return to one window, type ↑X1 .

Formatting Text To fill and justify text (make an even right margin), go to the beginning of the area to be formatted and type ↑@ , then go to the end of the area, move the cursor to the point in the line where you want the right margin to be and type ↑XF⇆1⇆Q (for a paragraph) or ↑XF⇆1⇆G (for a larger region).

Entering Control Characters as Text

If you need to enter a control character into your file as text, rather than typing it as a command, type ↑Q and then type the control character you are entering. The ↑Q will not appear on your screen. **Note:** This doesn't work for ↑C.

MISCELLANEOUS COMMANDS

↑O	Inserts blank line (above the line the cursor is at the beginning of), or breaks a line into two lines at the cursor
↑L	Clears screen and reprints it
↑T	Transposes characters before and at the cursor
⇆T	Transposes words before and at the cursor
⇆C	Changes first letter of word to uppercase (Don't type ↑C by mistake.)
⇆U	Changes word to uppercase
↑X↑U	Changes region to uppercase
⇆L	Changes word to lowercase
↑X↑L	Changes region to lowercase

8. SCRIBE -- BASIC COMMANDS

Scribe[7] is a text formatting program. It is used for preparing text to be a report, manual, article, letter, or any other type of document. Scribe processes text files that have formatting directions (commands) in them, and produces new files that can be printed on a variety of output devices. Some examples of the types of formatting commands are the following: italicize (make the following text appear in italics), center (center the following text), underline (underline the following text). Scribe can distinguish commands from text because every Scribe command starts with an at sign (@). The examples above would look like this: @i, @center, @u. Another Scribe command you should give indicates the output device (Penguin, Lineprinter, Diablo, or terminal screen) on which the file created by Scribe is to be printed. Once you use a text editor to create a file that contains text and Scribe commands, Scribe can process it and produce a second file, formatted according to the commands in the original file that you created.

First, use a text editor to create a file that has text along with Scribe commands. The Scribe commands are typed as if they were ordinary text. Then leave the text editor, return to the Exec level, enter Scribe, and give it the name of the file that you created. Scribe will read that file, interpret the commands, and create a second file by following the directions (commands) that you put in the first file. Now you may send the file that Scribe created to the appropriate output device.

Scribe is a large, complex program with many features that you may want to use. This chapter provides a general overview of Scribe. Much more detailed information can be found in the Scribe Introductory User's Manual by Brian Reid and Janet Walker.

[7] "Scribe" is a registered trademark of UNILOGIC, Ltd., Pittsburgh, PA.

8.1. FILENAME TYPES

The FileName Type is the part of the name that follows the first "." (period) in the name. With Scribe files, the Type is significant. **The file you create with Scribe commands in it should always have the Type .mss** (for example, *testfile*.mss). Scribe processes the .mss file you create and produces a second file with the same name but a different Type. The Type depends on the printing device you specified in the .mss file. The list below shows the file Type that Scribe associates with each printing device.

If Scribe detects any illogical commands while it's processing your .mss file, it will also produce an .err file, which contains a list of errors in the .mss file. When this happens, you must fix the problems in the .mss file yourself because **Scribe never alters the .mss file.** After you have fixed the .mss file, run it through Scribe again.

.mss Source file. When you create a file with Scribe commands in it, give it this Type. **This is the only file that you name or edit.**

.lpt Lineprinter file. When you use @Device[Lpt], Scribe will produce a file with the Type .lpt. The .lpt file must be printed on the Lineprinter.

.pod Diablo file. When you use @Device[Diablo], Scribe will produce a file with the Type .pod. The .pod file must be printed on the Diablo with the Podtyp program.

.prs Penguin file. When you use @Device[Penguin], Scribe will produce a file with the Type .prs. The .prs file must be printed on the Penguin.

.doc Terminal file. When you use @Device[File], Scribe will produce a file with the Type .doc. You can print this file on your terminal screen using the Type command at the Exec level.

.err Error file. If Scribe finds any errors while it is processing your .mss file, it prints them on your screen while it works and also produces an .err file that you can read with a text editor. Pay close attention to these error messages; they really mean what they say!

.otl Outline file. This file contains an outline of the corresponding .mss file's chapters, sections, and subsections. Scribe creates an .otl file only for certain kinds of documents (for example, articles and manuals).

8.2. GLOBAL COMMANDS

Global commands affect the entire file. If these commands are used, they must appear at the beginning of your .mss file. The three most often used Global commands are "Make," "Device," and "Style."

The @Make Command

If you use the @Make command, it should be the first line of your .mss file. This command indicates the kind of document you are creating and should be followed by one of the options shown enclosed in delimiters:

@Make[Text] If the @Make command is not given, it will default to @Make[Text]. This is the simplest document type and makes justified (even right margin) paragraphs.

@Make[Article] An Article has three levels of numbered headings: Section, Subsection, and Paragraph, as well as Appendix and AppendixSection.

@Make[Report] A Report has a title page, numbered chapters, sections, subsections, and a table of contents.

@Make[Manual] A Manual is like a Report, but can also have an index.

@Make[Slides] Slides uses a large font and appropriate line spacing for overhead transparencies.

@Make[Letterhead]

Letterhead produces a business letter format.
(Use @Make[ISILetterhead] for ISI stationery.)

The @Device Command

The second line of a Scribe .mss file should be the @Device command. If there is no @Make command, the @Device command should be the first line of the file. This command tells Scribe to create a file that can be printed on a specified device. It should be followed by one of the options shown, enclosed in delimiters:

@Device[File] This tells Scribe to create a .doc file. You may print a .doc file on your terminal screen with the Type command at the Exec level. If the @Device command is not given, Scribe defaults to @Device[File].

@Device[Lpt] This tells Scribe to create an .lpt file. The .lpt file should be printed on the Lineprinter with either the List or Print command at the Exec level.

@Device[Diablo] This tells Scribe to create a .pod file. This file should be printed on the Diablo with the Podtyp program (see page 62).

@Device[Penguin]
This tells Scribe to create a .prs file. This file should be printed on the Penguin with the Xpress command at the Exec level.

The @Style Command

Style commands allow you to change the appearance of the entire document. If they are used, Style commands should be grouped together immediately following the @Device command. Some useful Style commands are

@Style[Spacing 2] Makes double-spaced text.

@Style[Justification off] Makes filled text with unjustified right margin.

@Style[Indentation 0] Causes the first line of each paragraph to be flush left.

@Style[Indentation 5] Causes the first line of each paragraph to be indented five spaces.

8.3. USEFUL SCRIBE COMMANDS

Most commands can be in either of two forms:

1. The command followed by "delimiters" surrounding the text to be affected, for example, @center[Scribe Commands], or

2. The command delimited by "begin" and "end," for example,
   @begin[center]
   Scribe Commands
   @end[center]

The result of the two examples is the same:

Scribe Commands

Delimiters <u>must</u> match each other within a command. The delimiters you may use are: (...), [...], {...}, <...>, '...', and "..." . Choose your delimiters carefully so that the text or figures inside the delimiters do not include the particular delimiters used.

A blank line A blank line indicates the end of a paragraph.

@center This command centers text.

@i This command italicizes text.

@u This command underlines text. (See page 12 of the Scribe manual for underlining options.)

@verbatim Verbatim produces text that is not filled or justified but appears exactly as you typed it. The text in Verbatim is printed in a fixed-width font (similar to a typewriter). Other Scribe commands within Verbatim will still be recognized.

@format This command is similar to @Verbatim, but the text within @Format stays in the same font as the body of the text.

@quotation This command produces text that is single spaced, filled and justified, and with wider right and left margins (the margin is the white area on each side of the text).

@verse Verse produces results similar to @Quotation but does not fill and justify.

@itemize Itemize widens both the right and left margins, fills and justifies the text, and places a tick mark ("·") at the beginning of the first line of each new paragraph. Separate paragraphs as you type them with a blank line.

@enumerate Enumerate is almost the same as @Itemize except that numbers instead of tick marks are placed before each paragraph.

@description Description was used to format the text on this page. The word or phrase to be described is typed, followed by @\ (Scribe's tab indicator), followed by the description. Each word or phrase and its description are considered a paragraph, so separate paragraphs with a blank line.

@blankspace[n lines] or [n inches]
This command tells Scribe to insert blank lines in your text. Use @blankspace[5 lines] to insert 5 blank lines. Use @blankspace[3.4 inches] to insert a vertical space that size.

@ Use this to indicate a tab (instead of the tab key on your terminal). Be sure to use the \ (back slash) and not the / (forward slash). It is often necessary to use the Format or Verbatim command around the text with tabs so that the text is not filled and justified. (See page 56 for an example of using tabs.)

@> This command pushes whatever text follows it to the right margin (flush right). When @> is used with a tab, the text will be flushed against the tab instead of the right margin.

@ space bar Typing one @ sign followed by one space creates one and only one space in the text. Typing @ @ @ (three @ signs, each followed by a space) creates exactly three spaces in your text.

@@ Typing two @ signs together in the .mss file causes one @ sign to be printed in your text.

@newpage This command causes a page break.

@* This command causes a line break.

@group Group prevents the text from being broken between pages. If you have a paragraph or chart that is being broken up between pages in an awkward way, this command will cause it to be printed on one page. Usually this should be used in the @begin[group] ... @end[group] command format.

Remember to start all Scribe commands with an @ sign.

When using commands with @begin and @end, it is good to put the @begin[command] and @end[command] on lines by themselves. Every @begin must have an @end; otherwise Scribe will generate an error message.

Headings for documents with and without a table of contents are listed in the section on "Titles, Sections, and the Table of Contents" of the Scribe manual.

8.4. SENDING AN ".MSS" FILE THROUGH SCRIBE

Key:
↑ = Control Key
⇆ = Escape Key
↵ = Carriage Return

When your .mss file is ready to be sent through Scribe, type the following from the Exec level (underlined words are what you type):

```
@Scribe↵

Scribe 3B(1134)  Copyright (C) 1981  UNILOGIC, Ltd.
Default Database is <SCRIBE.3B>; Report bugs to SCRIBE.
*filename.mss↵

[Processing FILENAME.MSS.2
    [Device "PENGUIN"]
    [Document type "Text"
        [FontFamily Helvetica10]

[1  2  3]
4

**FILENAME.PRS for device PENGUIN has 4 pages.

@
```

In this example, Scribe processed the .mss file, prepared a .prs file for the Penguin, and returned to the Exec level. The .prs file can now be sent directly to the Penguin with the Xpress command.

To process a manuscript file through Scribe for a device other than the one specified by the @device command in the file, when typing in the FileName, follow it by a forward slash and the name of the device you would like (for example, *filename*.mss/lpt).

8.5. EXAMPLE OF .MSS AND .PRS FILE

Sample .mss file (see next page for the .prs version).

```
@device[Penguin]
@style[indentation 0]
@majorheading[Fun with Scribe]

The purpose of this file is to demonstrate the results of using
a few of Scribe's simple commands.
@i[Here is an example of the use of the italics command.]
@u[Here is an example of the use of the underline command], and
here is an example of the
use of the @center[center command.]
Look at this page to see the command or ".mss" file, and
look at the next page for the ".prs" file (the file Scribe
produced by following the commands in the .mss file).

The "itemize" and "enumerate" commands are also very simple
to use.
Use them with the @@Begin and @@End commands, and separate each
paragraph with a blank line.  The following was produced with
the @@enumerate command:
@begin[enumerate]
The only difference between "itemize" and "enumerate" is that
"enumerate" numbers each paragraph and "itemize" puts a tick
mark by each paragraph.

It is very important to remember to "end" any command that was
started with a "begin."  If you do forget, there will be at
least one error message when the file is run through Scribe.
@end[enumerate]
@blankspace[1 line]
@begin[description]
Example@\The description command separates a word or phrase
from text.  If the word or phrase overlaps the tab, the text
will start on the next line.  Separate paragraphs with a blank
line.
@end[description]
@blankspace[1 line]
@begin[quotation]
The Quotation command causes both margins to be indented,
fills and justifies the text, and
inserts a blank space before and after the text delimited by
the command.  The Verse command is similar, but does not
fill and justify.
@end[quotation]
```

.prs file

Fun with Scribe

The purpose of this file is to demonstrate the results of using a few of Scribe's simple commands. *Here is an example of the use of the italics command.* Here is an example of the use of the underline command, and here is an example of the use of the

center command.

Look at this page to see the command or ".mss" file, and look at the next page for the ".prs" file (the file Scribe produced by following the commands in the .mss file).

The "itemize" and "enumerate" commands are also very simple to use. Use them with the @Begin and @End commands, and separate each paragraph with a blank line. The following was produced with the @enumerate command:

1. The only difference between "itemize" and "enumerate" is that "enumerate" numbers each paragraph and "itemize" puts a tick mark by each paragraph.

2. It is very important to remember to "end" any command that was started with a "begin." If you do forget, there will be at least one error message when the file is run through Scribe.

Example The description command separates a word or phrase from text. If the word or phrase overlaps the tab, the text will start on the next line. Separate paragraphs with a blank line.

> The Quotation command causes both margins to be indented, fills and justifies the text, and inserts a blank space before and after the text delimited by the command. The Verse command is similar, but does not fill and justify.

8.6. TABS

Sample .mss file (see next page for .prs version).

```
@subsection[Setting Tabs]

@begin[verbatim]
@tabs[10,20,30,40]
@\344@\4859@\5938@\495
@\3988@\3975@\384@\394
@\@\@\397@\2987
@tabclear

@tabdivide[5]
@\234@\8796@\9876@\986
@\2368@\123@\6797@\23
@\37545@\2346@\3759@\234
@tabclear

@subsection[Tabs used with the Flushright Command]

@tabs[15,30,45]
@>$10100.00@\@>$2000.00@\@>$@ @ 3.00@\
@>10.00@\@>150.00@\@>130.00@\
@>320.00@\@>4500.00@\@>244.00@\

@tabclear
@tabdivide[4]
@>$10100.00@\@>$2000.00@\@>$@ @ 3.00@\
@>10.00@\@>150.00@\@>130.00@\
@>320.00@\@>4500.00@\@>244.00@\

@subsection[Setting Tabs Visually]

@tabclear
These people will arrive at 10:00 a.m.:  @↑Lisa Holt
@\Sheila Coyazo
@\Lori Holzer
@end[verbatim]
```

8.6.1. Setting Tabs

```
344       4859      5938      495
3988      3975      384       394
                    397       2987

   234          8796         9876         986
   2368         123          6797         23
   37545        2346         3759         234
```

8.6.2. Tabs used with the Flushright Command

```
$10100.00       $2000.00        $  3.00
    10.00         150.00         130.00
   320.00        4500.00         244.00

 $10100.00        $2000.00          $  3.00
     10.00          150.00           130.00
    320.00         4500.00           244.00
```

8.6.3. Setting Tabs Visually

```
These people will arrive at 10:00 a.m.:  Lisa Holt
                                         Sheila Coyazo
                                         Lori Holzer
```

8.7. SCRIBE PRACTICE

Homework #1

Follow these steps to Scribe

1. Use a text editor to create a file that contains some Scribe commands (try @center, @device, and @enumerate).

2. Exit from the editor naming the file *FileName*.mss. Always give a file containing Scribe commands the Type .mss.

3. You are now at the Exec level and ready to send the file you have created through Scribe. Type Scribe↲ and then type the name of the file you created (*FileName*.mss). Scribe processes that file by interpreting the commands and creates a new file with the same name but a different Type. The Type Scribe gives the file depends on the device command you used in your .mss file (Lpt, Penguin, File, or Diablo).

4. When finished, Scribe leaves you at the Exec level. Now you can give the Exec level command to send the file that Scribe produced to the appropriate output device. For the Lineprinter, type List *FileName*.lpt↲. For the Penguin, type Xpress *FileName*.prs↲ . For viewing on your terminal, type Type *FileName*.doc↲ .

5. Examine the output to see if you got the results you expected. If you didn't, check your .mss file to determine what went wrong. Correct your error, and then run the .mss file through Scribe again.

Homework # 2
Create this file

Practice Makes Perfect??

Use a text editor to create a file that has text along with Scribe commands. The Scribe commands are typed as if they were ordinary text. Then leave the text editor, return to Exec, enter Scribe, and give it the name of the file that you created. Scribe will read that file, interpret the commands, and create a second file by following the directions (commands) that you put in the first file. Now you may send the file that Scribe created to the appropriate output device. In this exercise, several commands will be used:

- @blankspace
- @itemize
- @description
- @u
- @i
- @center

It will be your HOMEWORK

TO USE THEM CORRECTLY!

Example The Description command creates an environment. This environment produces paragraphs with heading words in the margin. This environment is designed for reference tables containing a list of words or terms, each accompanied by a line or paragraph of description.

Second header The beginning of a new paragraph marks the beginning of a new description item.

Guess which
commands are
used here!!

THE END

Homework # 3
Create this file

Scribe the Easy Way

by I. Format Freely

July 8, 1980

Scribe is used to create formatted text. The file you create with a text editor contains text along with Scribe commands indicating how you would like the text to be formatted. Each Scribe command starts with the @ sign.

Most commands can be in either of two forms:

1. The command followed by "delimiters" surrounding the text to be affected, for example, @center[Scribe Commands], or

2. The command delimited by "begin" and "end," for example, @begin[center]
   Scribe Commands
   @end[center]

The result of the two examples is the same:

Scribe Commands

You can put a command into your manuscript file that tells Scribe which printing device your file is intended for. The following chart shows the command to use for each device.

@device[file]	creates .doc	file for the terminal screen
@device[penguin]	creates .prs	file for the Penguin
@device[diablo]	creates .pod	file for the Diablo
@device[lpt]	creates .lpt	file for the Lineprinter

It is not necessary to give a device command if your file is intended for the terminal screen. If no device command is given, the default is to create a file for the terminal screen.

8.8. USING SCRIBE FOR LETTERS

To create letters formatted for letterhead stationery, use @Make[Letterhead]. (Use @Make[IsiLetterhead] for ISI letterhead.)

Device options are:

- @Device[Diablo]
- @Device[Penguin]
- @Device[Lpt] Use this for proofreading a letter.
- @Device[File] Use this for proofreading a letter.

The command @String[Name = " "] (shown on page 63, in the sample .mss file) is useful only for letters with more than one page. After page one, the name given in the string (between the quotes) will appear at the top left of each succeeding page along with the page number in the center and the date on the right.

For paragraph indentation, use @Style[Indentation 5] (to indent 5 spaces) or @Style[Indentation 0] (for no indentation).

Penguin Letter If you have created a .prs file for the Penguin, you will use the Xpress command from the Exec level to send the file to the Penguin. After the file has been printed on the Penguin, you must photocopy it onto letterhead stationery.

Diablo Letter The default font for the Diablo is "Elite." If you plan to use the Elite print wheel, it is not necessary to have a font command in your .mss file. Be sure the pitch switch on the Diablo is set to 12. If you are going to use the Pica print wheel, use the command @Font[Pica] in your .mss file. For this print wheel the pitch switch should be set to 10. After Scribe produces the .pod file, use the Podtyp program on the Diablo to print your letter directly onto letterhead stationery.

8.9. THE PODTYP PROGRAM

A Scribe-created .pod file is printed on the Diablo using the Podtyp.Exe program. This summary shows a simple example of its use. Underlined words are what the user types. For more information on the use of these commands, and more complete documentation on the Podtyp program, see <Documentation>Podtyp.Doc.

```
@Podtyp.exe.27 ⇋)
B
Copyright (C) 1982 Bolt Beranek and Newman Inc.
PODTYP Version 2B(23)(Feb82).  See HELP PODTYP
Assuming XON/XOFF protocol.
Print file:  FileName.pod⇋
Preparing to print FileName.POD.1
12 pitch typewheel expected for this file

Commands are  B, C, D, E, F, L, N, P, R, S, W, X, and ?
Enter command: B
Begin processing.

Position paper and press the space bar
```

This example shows what you actually type and see printed on the Diablo. The user types (at the Exec level) Podtyp⇋. The Program types "Print File," to which the user responds with *FileName*.Pod⇋. The program then prints a list of optional commands. In this example, the "B" command was given to begin printing (you may also type ? to see a summary of the commands listed). The program types "Position paper and press the space bar" and then pauses until a fresh piece of paper is inserted and the space bar is pressed once; then it prints the file. The program pauses at the end of each page until the space bar is pressed. After the file has been printed, the user inserts a piece of scratch paper and then types a Q. The program types "Enter command:" and the user types D. The program stops and the user is left at the Exec level.

The Podtyp program is not limited to letters but can be used to print any .pod file (the file created by Scribe when you have @Device[Diablo] in your .mss file), or any text file.

Sample .Mss File for Penguin letter (see next page for .Prs version).

```
@Make[Letterhead]
@Device[Penguin]
@String[Name="Mr. Stevie Wonder"]
@Begin[Address]
Mr. Stevie Wonder
111 Super Street
Numberone, Everywhere  11111
@End[Address]
@Begin[Body]
@Greeting[Dear Stevie,]

Your last album, "Hotter than July," was truly hot and I loved
every minute of it.  "Rocket Love" was really spacy and all
the tracks were special in some way.  The problem is, I played
it sooo much, I'm sick of it and now I can't wait for your next
one.  Based on past experience, I'm afraid that you will keep
me in agony for another year and I just want you to know that
I will come unglued if you do!

I also want to know when you're going to give another concert
in Los Angeles.  All the kids here are saving their money for
that magic day when tickets for your concert go on sale and
they can wait in line all night to be the first to get them.

Don't forget, if you ever need an enthusiastic president for
your Stevie Wonder Fan Club, you know you can call on me.

@End[Body]
Yours forever,



Ms. Fan Ardent
@Begin[Notations]
FA/fa


cc:  @↑Mr. Peter Allen
@\Ms. Aretha Franklin

Enc.  "A Groupie's View of Stevie Wonder"
@End[Notations]
```

October 31, 1981

Mr. Stevie Wonder
111 Super Street
Numberone, Everywhere 11111

Dear Stevie,

Your last album, "Hotter than July," was truly hot and I loved every minute of it. "Rocket Love" was really spacy and all the tracks were special in some way. The problem is, I played it sooo much, I'm sick of it and now I can't wait for your next one. Based on past experience, I'm afraid that you will keep me in agony for another year and I just want you to know that I will come unglued if you do!

I also want to know when you're going to give another concert in Los Angeles. All the kids here are saving their money for that magic day when tickets for your concert go on sale and they can wait in line all night to be the first to get them.

Don't forget, if you ever need an enthusiastic president for your Stevie Wonder Fan Club, you know you can call on me.

Yours forever,

Ms. Fan Ardent

FA/fa

cc: Mr. Peter Allen
Ms. Aretha Franklin

Enc. "A Groupie's View of Stevie Wonder"

Suggested Reading

[ARPA 82] *ARPANET User's Guide*, Computing Analysis Corporation, Arlington, Va., 1982.

[DEC 80a] *TOPS-20 User's Guide*, Digital Equipment Corporation, Marlboro, Mass., 1980.

[DEC 80b] *Getting Started with Decsystem-20*, Digital Equipment Corporation, Marlboro, Mass., 1980.

[DEC 80c] *TOPS-20 Commands Reference Manual*, Digital Equipment Corporation, Marlboro, Mass., 1980.

[Holg 79] Holg, Chloe Sommers, *The Joy of TENEX and TOPS-20...in two parts*, USC/Information Sciences Institute, Marina del Rey, Calif., TM-79-15 and TM-79-16, January 1979.

[Mooers 82a] Mooers, Charlotte, *The Hermes Guide*, Bolt, Bernanek and Newman Inc., BBN Report 4995, 1982. Forthcoming.

[Mooers 82b] Mooers, Charlotte, *The Hermes Installation and Maintenance Guide*, Bolt, Bernanek and Newman Inc., BBN Report 4996, 1982. Forthcoming.

[Reid 80] Reid, Brian, and Janet H. Walker, *Scribe User's Manual*, Third edition, Unilogic, Ltd., Pittsburgh, Penn., 1980.

[Stallman 80] Stallman, Richard M., *EMACS Manual for TWENEX Users*, Massachusetts Institute of Technology, AI Memo 555, September 1980. Corresponds to EMACS version 150.

[XED 76] *XED User's Manual: Beginning Instruction*, USC/Information Sciences Institute, Marina del Rey, Calif., TM-76-3, May 1976.

Useful Addresses

Bolt Beranek and Newman Inc.
50 Moulton Street
Cambridge, Massachusetts 02238

Computing Analysis Corporation
1400 Wilson Blvd., Suite 1101
Arlington, Virginia 22209

Digital Equipment Corporation
P.O. Box CS-2008
Nashua, New Hampshire 03061

Document Distribution
USC/Information Sciences Institute
4676 Admiralty Way, Suite 1100
Marina del Rey, California 90291

Massachusetts Institute of Technology
Artificial Intelligence Laboratory
Attn: Publications
545 Technology Square
Cambridge, Massachusetts 02139

Unilogic, Ltd.
160 N. Craig
Pittsburgh, Pennsylvania 15213