RTE-M Editor

Reference Manual

HEWLETT hp PACKARD

HEWLETT-PACKARD COMPANY
11000 WOLFE ROAD, CUPERTINO, CALIFORNIA, 95014

Library Index Number
2RTEM.320.92064-90004

PART NO. 92064-90004
PRODUCT NO. 92064A

Printed in U.S.A. 1/77

LIST OF EFFECTIVE PAGES

Changed pages are identified by a change number adjacent to the page number. Changed information is indicated by a vertical line in the outer margin of the page. Original pages do not include a change number and are indicated as change number 0 on this page. Insert latest changed pages and destroy superseded pages.

Change 0 (Original) ..... Jan 1977

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

PREFACE

This publication is the reference manual for the RTE-M Editor (EDITM). EDITM is used to create and edit ASCII files. This manual provides a complete description of EDITM commands and functions. Operating system considerations are indicated, and the reader is referred to other RTE-M system manuals if additional information is required.

This manual is divided into five sections.

Section I — introduces EDITM, describes how to initiate interactive and batch jobs, surveys its operation, describes keyboard conventions and display formats, and describes a sequence of decisions to be made about the environment of EDITM.

Section II — explains the use of each command. Commands are grouped by function and examples are provided to show how each is used. Any special features or warnings are also provided.

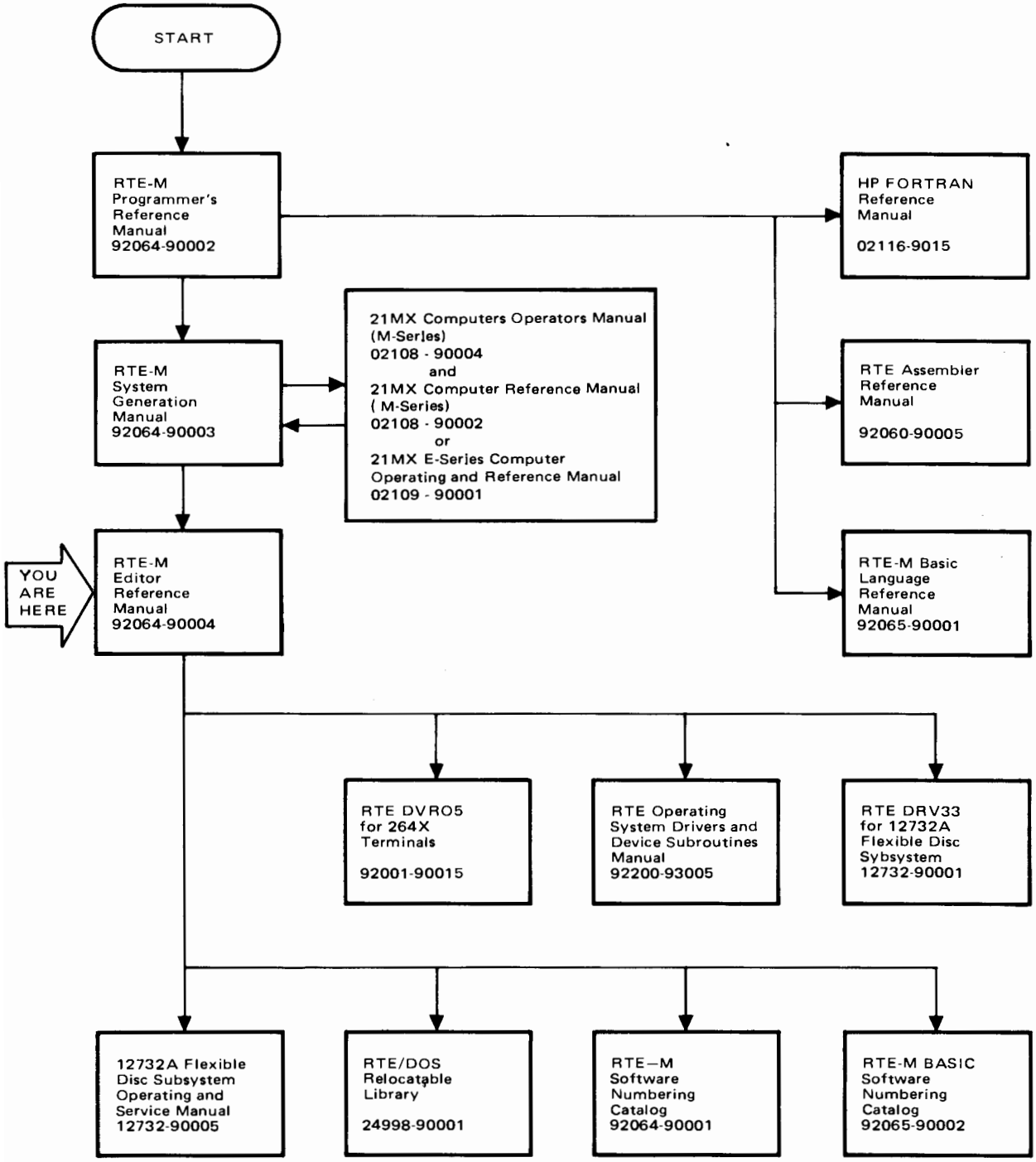
Section III —lists all EDITM messages, with causes and recovery procedures.

Section IV — provides an editing session example showing the commands used in combinations.

Section V — lists the special considerations for using EDITM in a batch environment.

The RTE-M Programmer's Reference Manual (92064-90002) should be available for reference when using this manual. Additional helpful information can be found in the publications listed on the RTE-M documentation map.

DOCUMENTATION MAP



CONTENTS

Section I Page

INTRODUCTION

EDITM Work Areas 1-1
Pending Line 1-2
Keyboard Conventions and Command Syntax 1-3
File Definition 1-4
Running EDITM 1-5
EDITM Termination 1-6
Display Format 1-6

Section II

EDITM COMMANDS

Program Listing 2-1
EDITM Command Summary 2-2
Control Commands 2-4
$ Command — Display or Change Special Characters 2-4
? Command — Display Command or Error Explanation 2-5
CTRL/G Command — Bell Control 2-6
T Command — Set Tab Character and Stops 2-7
W Command — Set Window 2-8
Command — Sequence Numbers 2-9
= Command — Set Line Length 2-10
K Command — Kill Trailing Blanks 2-11
M Command — Merge Source File 2-12
] Command — Line Terminator 2-13
Display Commands 2-14
P Command — Display the Pending Line 2-14
L Command — Display a Number of Lines 2-15
n Command — Display a Specified Line 2-16
/ or + Command — Space Down a Number of Lines 2-17
N Command — Display Pending Line Number 2-18
! Command — Display File Names 2-18
↑ or Command — Backup in the Destination Work Area 2-19
Line Edits 2-20
R Command — Replace Pending Line With Text 2-20
I Command — Insert Text Before Pending Line 2-21
Δ Command — Insert Text After Pending Line 2-22
– Command — Delete a Number of Lines 2-23
Character Edits 2-24
CTRL/R Command — Replace Characters 2-25
CTRL/I Command — Insert Characters 2-26
CTRL/C Command — Cancel Characters 2-27
CTRL/T Command — Truncate Characters 2-27
Character Edit Combinations 2-28
Character Edit Summary 2-28
Pattern Edits 2-28
Search Commands 2-29
B Command — Find a Line With a Find Field SOF to EOF 2-30
F Command — Find a Line With a Find Field From Pending Line to EOF 2-31
D Command — Delete Lines to Find Field or EOF 2-32
S Command — Merge Segment Following Pending Line 2-34
Exchange Commands 2-35
G Command — Character Replace on Pending Line 2-35
U Command — Unconditional Character Replace (No List) 2-36
V Command — Unconditional Character Replace (List) 2-38
X Command — Enable Exchange Pattern All Lines (List) 2-39
Y Command — Exchange on Pending Line, Display Next Occurrence of Pattern 2-40
Z Command — Enable Exchange Pattern All Lines (No List) 2-41
Terminate Commands 2-42
A Command — Abort EDITM 2-42
EC Command — End Edit and Create A File Manager File 2-43
ER Command — Replace Old File With New File 2-44
EN Command — End Edit and Rename Destination Work Area 2-46

Section III Page

EDIT MESSAGES 3-1

Section IV Page

SAMPLE EDIT 4-1

Section V Page

EDITM IN BATCH ENVIRONMENT 5-1

ILLUSTRATIONS

Title	Page	Title	Page
File/Work Area Relationship	1-1	EDITM Command File	5-1

TABLES

Title	Page	Title	Page
Command Syntax	1-3	Maximum Line Length for Common Devices	2-10
Keyboard Conventions	1-3	Character Edits Compared	2-28
EDITM Command Summary	2-2	EDITM Message Summary	3-1

INTRODUCTION

SECTION I

The RTE-M Editor (EDITM) is commonly used for:

- Creation of new programs or data files in ASCII code.
- Modification of new or existing ASCII code.
- Appending several ASCII files to each other to form a single file.

EDITM operates in either interactive or in batch mode. When used interactively, EDITM accepts operator commands from a keyboard device. When used in batch mode, commands are read from a File Manager file or an external device. Refer to Section V for batch mode operation.

1-1. EDITM WORK AREAS

EDITM references a source file and an output file and uses two temporary work areas (scratch files) on the disc. These two work areas are known as the source work area and the destination work area. Edited text is passed from the source file to the destination work area. The first editing pass is complete when EDITM has read and passed all of the text lines in the source file. Before another editing pass can be made, the destination work area becomes the source work area, and the destination work area is cleared. During the second and succeeding passes, the edited text is passed from the source work area to the destination work area. Upon completion of succeeding passes, the destination work area replaces the source work area and the destination work area is cleared for the next pass.

If you back up to edit a line preceding the current line that is beyond the limit of EDITM's memory buffering, the editing pass is completed. The remaining data in the source file or source work area is passed to the destination work area and the destination work area now becomes the source work area.

When editing is complete and EDITM is terminated with one of the terminate commands, the remaining data in the source file or source work area is passed to the destination work area, and the destination work area is written to an output file. The relationship between these files and work areas is shown in Figure 1-1.

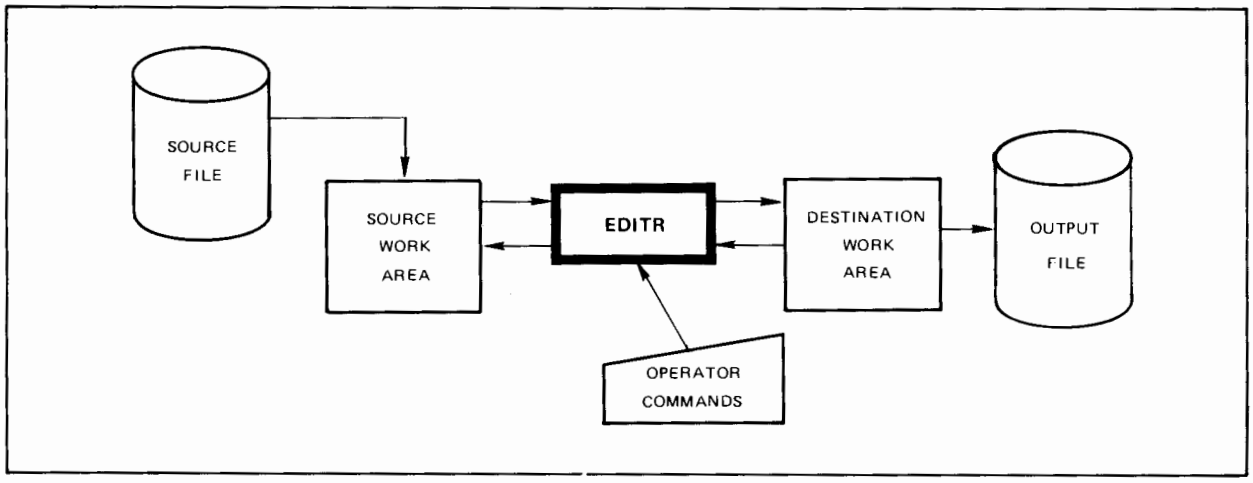


Figure 1-1. File/Work Area Relationship

1-2. PENDING LINE

When EDITM is run, the first line of the source file you name at the beginning of the editing session is read into the source work area and displayed. The displayed line is the current line available for editing and is called the "pending line". This line remains the pending line until you request a new pending line with an EDITM command. In this way you can continue to re-edit the same line until you are satisfied.

When you request a particular line of text, EDITM searches through the source work area until the requested line is encountered. That line then becomes the new "pending line" and is displayed on the session console. EDITM maintains a "pending line" pointer into the source area.

When the new pending line is displayed, the old pending line and all other lines passed over by the pointer in the search for the new pending line are usually written to the destination work area, and are no longer accessible until the "desintation work area" becomes the "source work area". In other words, you cannot go from line 0200 back to line 0100 without the source work area being replaced by the destination work area. This is called "physical rollover".

Generally you do not have to be concerned with this exchange because it is done automatically. However, you do need to remember that once the "source work area" is replaced, the previous "source work area" is totally gone, along with any lines deleted or changed in the last pass of EDITM. You should also remember that if you have made any insertions or deletions, these lines are no longer accessible by their original line numbers. For example, if line 3 is deleted in one editing pass, the original line 4 will be the new line 3. EDITM permits lines of text to be accessed without using line numbers.

EDITM does allow the user to back up part way in the destination work area without causing the "physical rollover" of the destination and source work areas. Part of these areas are buffered in memory and the pending line pointer may be backed up within the limits of this memory buffering without rolling over the work areas. This act is referred to as a "logical rollover". The limits of the logical rollover can be listed using the N command (see paragraph 2-18). Beyond this limit, physical rollover takes place.

1-3. KEYBOARD CONVENTIONS AND COMMAND SYNTAX

Keyboard commands are used to direct EDITM to do replacements, insertions, deletions, searches, and exchanges of text. These functions can be performed on characters within a line; they can be used to manipulate one entire line; or they can be used on groups of lines.

To use EDITM efficiently, you must know the command syntax and keyboard conventions it expects. They are shown and described in Tables 1-1 and 1-2.

Conventions Used in Examples are:

Δ Space entered at the keyboard.

(*x*) Circled lower case italic characters indicate a nonprinting control character entered at the keyboard.

Table 1-1. Command Syntax

CONVENTION	MEANING	COMMAND	EXAMPLE	COMMENTS
UPPER CASE BLOCK LETTERS	Literals that must be specified as shown.	**F**	**F**	No parameters
lower case italics	Variables to be replaced by values as defined in text.	**W***a,b*	**W7,9**	Values 7 and 9 replace variables a and b.
UPPER CASE ITALICS	Nonprinting control characters entered by pressing CTRL and a key simultaneously.	***CTRL/key***	***CTRL/C*** or ⓒ	In examples, control characters are indicated by circled key.
[ ]	Bracketed parameters are optional; if omitted default values are supplied.	file name [:security code [:cartridge reference number]]		
		Examples: MYFILE:AA:2 URFILE::−13		

Table 1-2. Keyboard Conventions

KEY	MEANING	FORMAT	EXAMPLE	COMMENTS
ESC	Escape key used in pattern edits to define field of indefinite length.	(esc) or @	F (esc) ASMB,R,L,T	Nonprinting character
RETURN	Carriage return used at end of line of input to transmit line to EDITM.	(CR)	/RNEW LINE (CR)	Nonprinting character
SPACE BAR	1. In place of a file name, causes EDITM to create an empty file. 2. As a command, inserts a line of text after pending line. 3. In text to indicate spaces.	Δ	SCR1? /Δ /ΔINSERT LINE /ΔMINΔΔΔBSSΔ1	Nonprinting
RUBOUT (TTY) or DELETE (2644/5)	Key used to delete line if pressed before RETURN key enters line; it causes line feed and carriage return to start of next line where correct line can be entered.	\	/RLINK \ RLINE (CR)	Prints as backslash; prompt is not repeated in next line
BACKSPACE or CTRL/A	Key used to backspace one character; effectively deletes one character each time key is pressed; some terminals use CTRL/H.		/RLINK E ^	May print as backspace or underscore

1-4. FILE DEFINITION

The File Manager parameter NAMR is used in EDITM commands to specify the files being used. This parameter is defined as a file name followed by two subparameters. The two subparameters may be omitted from the end of the list. If an embedded subparameter is omitted, its position must be indicated by the colon (:).

The format of the NAMR parameter is:

file name [:security code [:cartridge reference]]

Where:

file name
6-character ASCII file name; restricted as follows:

- only printable characters, space through _ (or ←)
- plus (+), minus (−), colon (:), or comma (,) not allowed
- first character must not be blank (space) or a number
- embedded blanks not allowed
- must be unique to FMP cartridge

security code
positive or negative integer or 2 ASCII characters representing a positive integer; range is from −32767 through 32767; security may be:

zero	file is unprotected (default)
+integer	write protected; may be read with any security or none; may be written only with correct code or negative (2's complement) of correct code.
−integer	file is fully protected; may be referenced only with correct negative code.

cartridge reference
positive or negative integer or 2 ASCII characters representing a positive integer; range is from −32767 through 32767; used to identify FMP disc cartridge, it may be:

zero	first available cartridge that satisfies the request is used; (default)
+integer	cartridge reference number (CR) by which the cartridge is identified.
−integer	logical unit number (LU) associated with the cartridge.

1-5. RUNNING EDITM

You can schedule EDITM from RTE-M with either the ON or RU operator command. The RTE-M Programmer's Reference Manual (29064-90002) contains the full explanation and implications of both commands.

The general form of the command to schedule EDITM is:

***ON**
***RU** **,EDITM** [*,lu,,* / *,fi,le,nm*] *,s*

Where:

lu
is the logical unit number of the device to be used for command input. The default for this parameter is the session console. Any device may be specified to enter EDITM commands; however, usually it is an interactive keyboard device. If you are editing in batch mode as described in Section V, this device could be a read-only device such as a card reader.

filenm
is the 6-character ASCII name of a File Manager file containing the command input.

s
0 to create scratch files
S to use existing files

When EDITM begins execution, it requires the names of the source file to be used and prompts you (if in interactive mode) for input by displaying a slash character (/).

EDITM requests the name of the source with:

```
INPUT?
```

After the prompt, the only legal responses you may enter are the name of an existing file (refer to File Definition for the format) and a space. Any other response will cause a **"FILE NOT FOUND"** error message from the File Manager. If you desire to abort EDITM at this time, you may enter **CTRL/D** (end-of-file).

If you enter S for the s parameter in the run command, EDITM will prompt you for the names of the scratch files with:

```
SCR1?
SCR2?
```

The responses for these prompts follow the same rules as stated above.

If 0 was specified for the s parameter in the run command, EDITM will create a 30 block, type 3 scratch file on logical unit 2. These files will be automatically purged at the completion of the edit session. When a space is entered as a response to **"INPUT?"**, the empty file specified as the second scratch file will be used as the input file.

If a legal file name is used as a response for the source file, EDITM obtains the file from the File Manager and displays the first line of the file. This is the pending line and the editing can now begin.

1-6. EDITM TERMINATION

The EDITM terminate commands assign the final version of text in the destination area to a File Manager type 4 file. The destination area can be output directly to a device through a type 0 file or by specifying an lu number.

1-7. DISPLAY FORMAT

The pending line displayed and listings output by EDITM are always preceded by two blanks. This convention allows room for the EDITM prompt and a single character command, and results in your new text being automatically aligned with that displayed by EDITM.

EDITM COMMANDS

SECTION II

2-1. PROGRAM LISTING

A complete listing of the program (TIME) used in the command examples is provided here for reference. The example was produced by using the FMGR :LI command so that the file name and line numbers can be used for correlation.

```
TIME  T=00004 IS ON CR00002 USING 00003 BLKS R=0024

0001 ASMB,R,L,T
0002        NAM TIME
0003        ENT WEN
0004        EXT .ENTR,EXEC
0005 HRS    BSS 1
0006 MIN    BSS 1
0007 SEC    BSS 1
0008 WEN    NOP
0009        JSB .ENTR
0010        DEF HRS
0011        JSB EXEC
0012        DEF *+3
0013        DEF TCODE
0014        DEF TBUFF
0015        LDA TBUFF+3
0016        STA HRS,I
0017        LDA TBUFF+2
0018        STA MIN,I
0019        LDA TBUFF+1
0020        STA SEC,I
0021        JMP WEN,I
0022 TCODE  DEC 11
0023 TBUFF  BSS 5
0024        END
```

2-2. EDITM COMMAND SUMMARY

All of the EDITM commands are summarized in Table 2-1. This table presents the commands in the same functional groups in which they are described in this section.

Table 2-1. EDITM Command Summary

FUNCTION	COMMAND	DESCRIPTION	PAGE NO.
CONTROL	$	Change or Display Special Characters	2-4
	?	Display Command or Error Explanation	2-5
	CTRL/G	Bell Control	2-6
	T	Set Tab Character and Stops	2-7
	W	Set Window	2-8
	#	Sequence Numbers	2-9
	=	Set Line Length	2-10
	K	Kill Trailing Blanks	2-11
	M	Merge Source File Following Pending Line	2-12
	]	Line Terminator	2-13
DISPLAY	P	Display Pending Line	2-14
	L	Display a Number of Lines	2-15
	n	Display a Specified Line	2-16
	/ or +	Space Down a Number of Lines and Display	2-17
	N	Display Pending Line Number, Approximate Number of Words in the Destination Work Area, Number of Characters in the Pending Line, and Backup Limit	2-18
	↑ or ^	Backup in Destination Work Area	2-19
	!	Display Names of All Files	2-18
LINE EDITS	R	Replace Pending Line With Text	2-20
	I	Insert Text Before Pending Line	2-21
	Δ	Insert Text After Pending Line	2-22
	–	Delete a Number of Lines	2-23
CHARACTER EDITS	P	Edit Pending Line	2-24
	C	Edit Pending Line and Advance to Next Line	2-24
	O	Make a Copy of Pending Line and Edit	2-24
	CTRL/R	Replace Characters	2-25
	CTRL/I	Insert Characters (CTRL/S	2-26
	CTRL/S	Alternate for CTRL/I)	2-26
	CTRL/C	Cancel Characters	2-27
	CTRL/T	Truncate Remainder of Pending Line	2-27
PATTERN EDITS	Search Commands		
	;	Find Field Tabbed	2-29
	esc or @	Find Field of Indefinite Length	2-29
	/	Find Field Within Window	2-29
	CTRL/@	Find a Zero Length Line With This Field	2-29
	B	Find a Line With Find Field — SOF to EOF	2-30
	D	Delete Lines to Find Field or EOF	2-32
	F	Find a Line With a Find Field From Pending Line to EOF	2-31
	S	Merge Segment Following Pending Line	2-34

Table 2-1. EDITM Command Summary (Continued)

FUNCTION	COMMAND	DESCRIPTION	PAGE NO.
	Exchange Commands		
	G	Character Replace on Pending Line	2-35
	U	Unconditional Exchange (no list)	2-36
	V	Unconditional Exchange (list)	2-38
	X	Enable Exchange Pattern All Lines (list)	2-39
	Y	Exchange on Pending Line, Display Next Occurrence of Pattern	2-40
	Z	Enable Exchange Pattern for Next Edit (no list)	2-41
TERMINATE	A	Abort EDITM	2-42
	EC	Create File Manager File	2-43
	ER	Replace Old File With New File	2-44
	EN	Rename Destination Scratch File	2-46

2-3. CONTROL COMMANDS

The control commands provide parameters for EDITM features you may desire in addition to the normal text manipulation functions.

2-4. $ COMMAND — DISPLAY OR CHANGE SPECIAL CHARACTERS

There are five characters that have a special meaning in EDITM commands. These are the tab character (;), the indefinite find field character (esc or @), the delimiter character (/), the S command delimiter (–). and the line terminator character (]). The $ command allows you to change these characters in order to use them as part of the text. The $ without parameters will display the current values of these characters.

Format:

$ [*xy* [,xy [. . .]]]

Where:

x
may be one of the following:

T for the tab character

E for the indefinite find field character

D for the delimiter character

S for the S command delimiter

L for the line terminator character

y
is the new control character for the function specified by *x*.

Examples:

Display the current values.

```
/$
  T=; D=/ E=@ S=- L=]
```

Change the tab character to @.

```
/$T@
   T=@ D=/ E=@ S=- L=]
```

2-5. ? COMMAND — DISPLAY COMMAND OR ERROR EXPLANATION

The ? command allows the user to obtain an explanation of a command or error message if the disc-based Help File is available.

Format:

?	Display a summary of all EDITM commands and errors
?*x*	Display an explanation of the command or error specified
?,*lu* **?*x*,*lu***	Write the explanation to a list device
?,*name* **?*x*,*name***	Write the explanation to a list file

Where:

x
is a command or error number.

lu
is the logical unit number of a list device.

name
is *filename*[*:security code*[*:cartridge reference number*]]. Refer to File Definition for parameter descriptions. If the file has been created with a write protect *security code* this must be specified; *cartridge reference number* should be specified if there is a possibility of another file of the same name on another cartridge; otherwise, *filename* is sufficient.

Example:

Display a summary of all EDITM commands and error messages.

```
/?
                    COMMAND/ERROR SUMMARY

A-ABORT              T-SET TABS            0-INVALID PARAMETER
B-SEARCH FROM START  U-EXCHANGE IN WINDOW  1-INVALID COMMAND
C-EDIT PL            V-SAME AS "U" W/LIST  2-COMMAND FILE NOT FOUND
D-DELETE TO PATTERN  W-SET WINDOW          3-FILE TOO LARGE
E-EDIT EDITM         X-SAME AS "Z" W/LIST  4-DELIMITER MISSING
F-FIND PATTERN       Y-EXCHANGE & FIND     5-NO ROOM
G-EXCHANGE IN PL     Z-ENABLE EXCHANGE     6-PARAMETER MISSING
I-INSERT BEFORE PL   #-ADD LINE NUMBERS    7-DISC FULL
K-TRAILING BLANKS    +-ADVANCE N LINES
L-LIST N LINES       /-ADVANCE N LINES
M-MERGE FILE         DELETE N LINES
N-LIST STATUS         -INSERT AFTER PL
O-COPY & EDIT PL      -BACK UP N LINES
P-EDIT PL            !-LIST FILE NAMES
R-REPLACE PL         $-LIST SPECIAL CHARS
S-SEARCH AND MERGE   =-SET LINE LENGTH     'PL': PENDING LINE

? FOLLOWED BY ANY CHARACTER LISTS MORE DETAILS.
```

Display the explanation of the A command.

```
/?A
THE 'A' COMMAND CAUSES THE EDITOR TO ABORT. ALL FILES
REMAIN UNCHANGED.
```

Display the explanation of the EDITM-3 error message.

```
/?3
EDITM 3-FILE TOO LARGE
THIS COMMAND REQUIRES LESS THAN 32000 RECORDS AND
 LESS THAN 64000 WORDS IN THE FILE.
```

2-6. CTRL/G COMMAND — BELL CONTROL

When EDITM is scheduled on a terminal with a bell, the bell is rung automatically every time a prompt is displayed. This command is used to turn the bell OFF; or, if OFF, to turn it back ON. The control character is nonprinting.

Format:

CTRL/G

2-7. T COMMAND — SET TAB CHARACTER AND STOPS

You may change the tab character and the standard tab stops with the T command. When EDITM is turned on, the initial tab character is the semicolon (;). The stops are initially set for the 7th and 21st columns.

Format:

T	disable tabs.
T*x*	Change tab control character, leave stops.
T*xs1,s2, . . .,s10*	Change tab stops and control character.

Where:

x
is the new tab stop control character (replaces the original semicolon or current tab control character).

s1,s2, . . . ,s10
are the new column numbers of the stops replacing 7 and 21. Tabs used beyond the highest defined stop are replaced with blanks. The tab character may be changed without changing the stops. The maximum number of tab stops that can be defined at one time is 10.

Example:

Change the tab character to a percentage sign (%) without changing the stops.

```
/T%
   T=% D=/ E=@ S=- L=]
```

Change the tab stops to columns 4 and 9 and leave the tab character

```
/T;4,9
   T=; D=/ E=@ S=- L=]
```

2-8. W COMMAND — SET WINDOW

When the Editor is turned on, the display field or "window" consists of columns 1 through 150. This command resets the window to new boundaries. It is particularly useful with the search commands (B, D, F, and S) and the exchange commands (G, U, V, X, Y, and Z) to limit the area of each record which is searched.

Format:

W*a,b*

Where:

a
is the initial column number (default is 1).

b
is the final column number (default is 150).

Example:

Set the window to include only columns 7, 8, and 9.

```
/W7,9
```

2-9. #COMMAND — SEQUENCE NUMBERS

You can use this command to cause EDITM to put sequence numbers on all lines in the file. A three-character identifier begins in column 73 and the numbers begin in column 76.

Format:

#	Start numbers in column 76.
#xxx	Start numbers in column 76 preceded by *xxx*.
#xxxnumb1	Start numbers with *numb1* value. *xxx* field required.
#xxxnumb1,numb2	Start numbers with *numb1* value, increment by *numb2*. *xxx* field required.

Where:

xxx
is the three-character identifier. It occupies columns 73-75 and must be included when specifying *numb1* and *numb2* (it may contain blanks, however).

numb1
is the starting number. The first line will start with this number. If omitted, numbers start with 00000.

numb2
is the incrementing value. *numb1* is incremented by *numb2* for each line. If omitted, *numb1* is incremented by 10.

Example:

Put sequence numbers on all lines in the file using the default parameters.

```
/#
  EOF
```

Start sequence numbers in column 76 with 1, increment by 1 and precede the sequence numbers with the character string 001.

```
/#0011,1
  EOF
```

2-10. = COMMAND — SET LINE LENGTH

The line length initially defaults to 150 characters when EDITM is turned on. This command allows you to reset the line length to another value and to truncate any characters in the line beyond the new limit.

Format:

```
=n
```

Example:

Request line number 5 and set the line length to five characters. Display the pending line to show the results of the edit.

```
/5
  HRS    BSS 1
/=5
/5
  HRS
```

You can specify a line length that is compatible with the device your output file is going to. Otherwise, long lines may be truncated. Table 2-2 shows the output devices commonly used with EDITM and the number of characters per line each supports. You should remember that displayed information is preceded by two spaces.

Table 2-2. Maximum Line Length for Common Devices

DEVICE	LINE LENGTH	COMMENTS
Punched Cards	80 Characters	
CRT	72-80 Characters	
Magnetic Tape	150 Characters	Default maximum length supported by EDITM.
TTY Device	72 Characters	Longer lines will encounter printing problems.
Paper Tape	150 Characters	Default maximum length supported by EDITM.
Disc	150 Characters	Default maximum length supported by EDITM.
Line Printer	80-132 Characters	Number of characters varies with printer model. Consult appropriate manual for your printer.

2-11. K COMMAND — KILL TRAILING BLANKS

This command deletes trailing blanks from all lines in the work area. In addition, the command always causes the source work area to be replaced by the destination work area, and always ends with an EOF message.

Format:

```
K
```

Example:

Pad pending line with trailing blanks and display the number of characters in the pending line. The character count includes the blanks. Kill the trailing blanks in the file and redisplay the line that had been padded. The character count now reflects the edit.

```
/RSEC    BSS 1
/N
  N=00007  =00006
  W=00041 C=00015
/K
  EOF
/7
  SEC    BSS 1
/N
  N=00007  =00006
  W=00041 C=00012
```

The initial character count includes a padded blank in the example below. Character-by-character replacement, omitting the padded blank shows the actual character count. Kill the trailing blanks in the file and redisplay the line. EDITM always makes the number of characters in the record even when writing lines to its destination work area.

```
/9
         JSB .ENTR
/N
   N=00009  =00008
   W=00052 C=00016
/R       JSB .ENTR
/N
   N=00009  =00008
   W=00052 C=00015
/K
  EOF
/9
         JSB .ENTR
/N
   N=00009  =00008
   W=00052 C=00016
```

2-12. M COMMAND — MERGE SOURCE FILE

This command is used to include any part of a file within the file in the destination work area. The merge takes place after the pending line and before the next line. The next line becomes the new pending line. This command may be used to correct an accidental delete by specifying the source work file as the merge file and merging the deleted lines. If the file name is not specified, the source work area is assumed.

Format:

M*name,first,last*
M*lu,first,last*

Where:

name
is *filename*[*:security code*[*:cartridge reference number*]]. Refer to File Definition for parameter descriptions. If the source file has been created with a read protect *security code*, this must be specified; *cartridge reference number* should be specified if there is a possibility of another file of the same name on another cartridge, otherwise; *filename* is sufficient.

first
is the first line to be merged (default is 1).

last
is the last line to be merged (default is the entire file).

Example:

Merge the file named PRINT after the pending line.

```
/MPRINT
```

Merge lines 5, 6, 7, 8, 9, and 10 of the source work area after the pending line.

```
/M,5,10
```

2-13.] COMMAND — LINE TERMINATOR

The] command allows you to enter comments following another EDITM command. This command is sometimes necessary in batch mode to eliminate trailing blanks in the command file.

Format:

command text] *comments*

Where:

command
is any EDITM command.

text
is EDITM text used with appropriate EDITM commands.

comments
are any comments you may wish to enter.

Example:

```
/$T@] CHANGE THE TAB CHARACTER TO @
```

2-14. DISPLAY COMMANDS

These commands cause the contents of the source work area, or information about the contents to be displayed. The display normally occurs on the session terminal, but some commands allow displays on the line printer, the console, etc.

2-15. P COMMAND — DISPLAY THE PENDING LINE

This command displays the pending line on the session console. You may then modify it if you wish. The P command can also be used in character edits as described in paragraph 2-27.

Format:

```
P
```

Example:

Display the pending line.

```
/P
  ASMB,R,L,T
```

2-16. L COMMAND — DISPLAY A NUMBER OF LINES

This command displays the pending line, plus a specified number of lines on a specified device or list file. The list device must be an output device such as a terminal, line printer, punch, etc. If it is not an output device, EDITM will continue to run and the source area pointer will be advanced, but error messages are displayed. It should be noted that two blanks are inserted as the first two characters of each line displayed, and that the new pending line displayed on the session console is the last line listed.

Format:

L*n*	List *n* lines
L*n*,*lu*	List *n* lines on logical unit *lu*
L*n*,*name*	List *n* lines to file name

Where:

n
is the number of lines to be printed.

lu
is the optional logical unit number of the list device. The default is the session console.

name
is *filename*[*:security code*[*:cartridge reference number*]]. Refer to File Definition for parameter descriptions. If the source file has been created with a write protect *security code*, this must be specified; *cartridge reference number* should be specified if there is a possibility of another file of the same name on another cartridge; otherwise, *filename* is sufficient.

Example:

List the pending line plus five more lines.

```
/L5
  ASMB,R,L,T
        NAM TIME
        ENT WEN
        EXT .ENTR,EXEC
  HRS   BSS 1
  MIN   BSS 1
```

2-17. *n* COMMAND — DISPLAY A SPECIFIED LINE

This command displays the requested line and makes it the pending line. If the line number is less than or equal to the current pending line number, the destination work area replaces the source work area, and changes the text line numbers if any insertions or deletions were made to the text.

Format:

n	Display line *n*.
n,lu	Write line *n* on logical unit lu.
n,name	Write line *n* to file *name*.

Where:

n
is the line number.

lu
is the optional logical unit number of the list device. The default is the session console.

name
is *filename*[*:security code*[*:cartridge reference number*]]. Refer to File Definition for parameter descriptions. If the source file has been created with a write protect *security code*, this must be specified; *cartridge reference number* should be specified if there is a possibility of another file of the same name on another cartridge; otherwise, *filename* is sufficient.

Example:

Display lines 12, 13, and 14. Reposition to line 12 and insert two lines and redisplay the lines. It is assumed in the example that the first line number is 1, the second line number is 2, etc. A zero (0) is interpreted as 1.

```
/12
      DEF *+3
/13
      DEF TCODE
/14
      DEF TBUFF
/12
      DEF *+3
/     DEF *5
/     DEF *+2
/12
      DEF *+3
/13
      DEF *5
/14
      DEF *+2
/15
      DEF TCODE
```

2-18. / OR + COMMAND — SPACE DOWN A NUMBER OF LINES

This command advances the pending line the specified number and displays the new pending line.

Format:

/	or	+	Advances the pending line by 1.
/n	or	*+n*	Advances the pending line the number specified.
/n,lu	or	*+n,lu*	Advances the number of lines specified and displays the new pending line at the specified *lu*.
/n,name	or	*+n,name*	Advances the number of lines specified and writes the new pending line to the specified file.

Where:

n
is the number of lines to be skipped. The default is 1.

lu
defines the logical unit number of the device on which the display occurs. The default is the session console.

name
is *filename*[*:security code*[*:cartridge reference number*]]. Refer to File Definition for parameter descriptions. If the source file has been created with a write protect *security code*, this must be specified; *cartridge reference number* should be specified if there is a possibility of another file of the same name on another cartridge; otherwise, *filename* is sufficient.

Example:

```
/1
  ASMB,R,L,T
/+5
  MIN    BSS 1
/1
  ASMB,R,L,T
//5
  MIN    BSS 1
/1
  ASMB,R,L,T
/+
         NAM TIME
//
         ENT WEN
```

To skip over a line of text use +2 or /2 with the results:

LINE 1 — Pending line when +2 entered
LINE 2 — Skipped
LINE 3 — New pending line

2-19. N COMMAND — DISPLAY PENDING LINE NUMBER

This command displays the number of the pending line in the source work area (N), the backup limit for the ↑ command (↑), the number of words in the destination work area (W), and the number of characters in the pending line (C).

Format:

```
N
```

Example:

```
/1
  ASMB,R,L,T
/+3
        EXT .ENTR,EXEC
/N
  N=00004  =00003
  W=00019 C=00020
```

2-20. ! COMMAND — DISPLAY FILE NAMES

This command prints the names of the original source file, the source work file, and the destination work file.

Format:

!	Display all file names.
!*lu*	Display all file names on logical unit *lu*.
!*name*	List all file names to file *name*.

Where:

lu
is the logical unit number of the list device.

name
is *filename*[*:security code*[*:cartridge reference number*]]. Refer to File Definition for parameter descriptions. If the source file has been created with a write protect *security code*, this must be specified; *cartridge reference number* should be specified if there is a possibility of another file of the same name on another cartridge; otherwise, *filename* is sufficient.

Example:

```
/!
  CAROL
  INFILE= CAROL OUTFILE= 1EDITR
```

2-21. ↑ COMMAND — BACK UP IN THE DESTINATION WORK AREA

This command allows you to edit lines in the destination work area. It does this by causing the pointer in the destination work area to back up. The source and destination work area are buffered in memory. This command moves lines from the destination work area to the source work area. The N command displays the backup limit of this memory buffering.

An up arrow (↑) without a parameter backs up one line. It should be noted that the up arrow command forces a transfer from the source work area to the destination work area if the backup limit is exceeded.

Format:

↑*n* or ^*n*

Where:

n
is the number of lines to back up in the destination work area.

Example:

Display line 5 then backup 2 lines and display the new pending line.

```
/5
  HRS    BSS 1
/↑2
         ENT WEN
```

If *n* is so large that it causes the pointer to back up past the beginning of the destination work area, the first line of the destination file becomes the pending line.

2-22. LINE EDITS

You can manipulate one entire line of text by doing line edits. EDITM provides four commands (R, I, Δ, and O) to replace, insert, and duplicate a line of text, plus one command to delete a line of text (−*n*).

2-23. R COMMAND — REPLACE PENDING LINE WITH TEXT

This command replaces the pending line with new text entered at the session console. If the command is used with no new text input, the pending line becomes zero length.

Format:

R*text*

Example:

Make line 6 the pending line and display it. Replace it with the new line and redisplay line 6.

```
/6
  MIN    BSS 1
/RMAX    BSS 5
/6
  MAX    BSS 5
```

2-24. I COMMAND — INSERT TEXT BEFORE PENDING LINE

The I command inserts a new line of text before the pending line. If this command is entered without new text, the inserted line becomes zero length.

Format:

Itext

Example:

List 5 lines of text and then move the pending line pointer to line 5. Insert a new line of text. Move the pointer back to the beginning of the file and display 5 lines of text again.

```
/L5
  ASMB,R,L,T
         NAM TIME
         ENT WEN
         EXT .ENTR,EXEC
  HRS    BSS 1
  MIN    BSS 1
/5
  HRS    BSS 1
/ISHR    BSS 1
/1
  ASMB,R,L,T
/L5
  ASMB,R,L,T
         NAM TIME
         ENT WEN
         EXT .ENTR,EXEC
  SHR    BSS 1
  HRS    BSS 1
```

2-25. Δ COMMAND — INSERT TEXT AFTER PENDING LINE

This command is indicated by a space and is used to insert new text immediately after the pending line. If the command is used without new text, the new line has zero length. The new line becomes the pending line.

Format:

Δ*text*

Example:

Make line 4 the pending line and insert a new line after it. Go back to line 4 and verify the insertion by listing three lines.

```
/4
        EXT .ENTR,EXEC
/ HHR   BSS 1
/4
        EXT .ENTR,EXEC
/L2
        EXT .ENTR,EXEC
  HHR   BSS 1
  HRS   BSS 1
```

2-26. – COMMAND — DELETE A NUMBER OF LINES

This command deletes a specified number of lines in the file. If no value is specified, one line is deleted.

Format:

–n

Example:

List the first 10 lines and then reposition the pending line pointer to line 4. Delete 5 lines including line 4. Reposition the pending line pointer to the beginning of the file and list the first 10 lines of the file again.

```
/L9
  ASMB,R,L,T
        NAM TIME
        ENT WEN
        EXT .ENTR,EXEC
  HRS   BSS 1
  MIN   BSS 1
  SEC   BSS 1
  WEN   NOP
        JSB .ENTR
        DEF HRS
/4
        EXT .ENTR,EXEC
/-5
        JSB .ENTR
/1
  ASMB,R,L,T
/L9
  ASMB,R,L,T
        NAM TIME
        ENT WEN
        JSB .ENTR
        DEF HRS
        JSB EXEC
        DEF *+3
        DEF TCODE
        DEF TBUFF
        LDA TBUFF+3
```

2-27. CHARACTER EDITS

The character edit commands provide a means to change the contents and modify the structure of the current line of text. Four commands allow the replacement, insertion, and deletion of characters. Each of these commands uses nonprinting control characters so that character alignment is maintained in the text. Each must be used with an initial "P", "C", or "O" command.

The current delimiter is used as a "place holder" to preserve existing text in the pending line. The P, C, and O commands are not themselves character edit functions. They are used to determine the line disposition during the edit: P leaves the edited line as the pending line, C advances the pending line to the next line after the edit, O sends the pending line to the destination work area and then edits a copy of the line and leaves it pending.

When C is entered as the first character:

- the pending line is edited
- the results of the edit are displayed
- the edited line is passed to the destination work area
- the next line of code is displayed and is the new pending line.

When P is entered as the first character:

- the pending line is edited
- the results of the edit are displayed.
- the edited line remains the pending line, allowing you to make further edits to this line.

When O is entered as the first character:

- the pending line is passed to the destination work area
- a copy of the pending line is retained and is edited
- the results of the edit are displayed
- the edited line is retained as the pending line.

2-28. CTRL/R COMMAND — REPLACE CHARACTERS

Character replacement allows you to change any character in the pending line. Any embedded character you wish to preserve in the original text is skipped by entering the current delimiter. Using the tab character will also cause characters to be skipped. Skipped characters in the new line appear exactly as they did in the old line. A carriage return preserves the rest of the line unless CTRL/T was specified to truncate.

Character replacement is the EDITM default mode. This means the control character CTRL/R can be omitted when EDITM is initially turned on, or following a P, C, or O command. It is only required following one of the control characters CTRL/I, CTRL/S, or CTRL/C.

Format:

```
P CTRL/R
C CTRL/R
O CTRL/R
```

Example:

Replace characters in the pending line and leave the current line as the pending line. Preserve original characters with the current delimiter.

```
/P
  MIN    BSS 1
/P//////JSB
  MIN    JSB 1
```

Tab to column 7 (skip) and change operation field. Display the edit result. No control character is required since EDITM is still in default mode.

```
/P
         DEF TCODE
/O;JSB
         JSB TCODE
```

Add ABC to the end of the pending line and display the next line making it the new pending line.

```
/P
         DEF *+3
/C//////////////ABC
         DEF TCODE
```

2-29. CTRL/I COMMAND — INSERT CHARACTERS

This command is used to insert characters in the pending line. There are two formats for this command: CTRL/I and CTRL/S. CTRL/I is the standard format. CTRL/S is used on terminals where CTRL/I has a special function (e.g., the HP 2754 teleprinter uses CTRL/I as a tab function). Each new character is inserted in the line immediately before the character under which the CTRL/I or CTRL/S is entered. Once the control character is entered, entering the current delimiter to do character skipping inserts blanks. The tab character also causes the tabbed number of blanks to be inserted.

Format:

P CTRL/I		**P CTRL/S**
O CTRL/I	or	**O CTRL/S**
C CTRL/I		**C CTRL/S**

Example:

Insert the word LABEL in the pending line starting in column 1.

```
/P
        NOP
/PLABEL
 /LABEL NOP
(I)
```

Note that CTRL/I or CTRL/S is a nonprinting character and appears in the example with a circle around it for clarity. Since characters cannot be skipped in the Insert Mode, all skipping must first be done in the Replace Mode, before changing the mode to Insert. To insert a letter in a word, as in the example below, the position is aligned in the Replace Mode, and the mode is then changed to Insert.

```
/P
  BUFER DEC 6
//
        DEF BUFR
/P/////////////E
               ↑
               └──(I)

        DEF BUFER
```

2-30. CTRL/C COMMAND — CANCEL CHARACTERS

This command is used to delete characters from the pending line. Each character or place holder entered after CTRL/C will delete one character from the pending line. Character skipping (i.e., entering the current delimiter) deletes characters. The tab character deletes everything up to the tab stop. When carriage return is entered, the pending line will be left-justified.

Format:

```
P CTRL/C
O CTRL/C
C CTRL/C
```

Example:

Edit line to correct spelling error in comment. The tabs are assumed in columns 7 and 21. The X is used as a place holder for the S to be deleted.

```
/        SSA,RSS       SSKIP IF NEG
/P;;X
          SSA,RSS       SKIP IF NEG
          (C)
```

2-31. CTRL/T COMMAND — TRUNCATE CHARACTERS

The CTRL/T command is used to delete characters from the end of the pending line. When the control character is entered, the remainder of the line is eliminated.

Format:

```
P CTRL/T
O CTRL/T
C CTRL/T
```

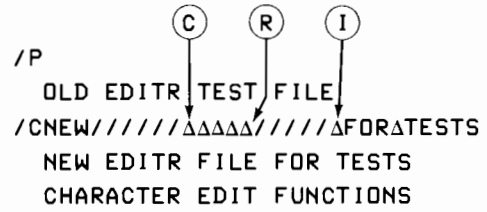
Example:

Delete DEF from the end of the pending line.

```
/P
  ABCDEF
/P///
  ABC
    (T)
```

2-32. CHARACTER EDIT COMBINATIONS

Several types of character editing can be done on one line of text. As an example; Replace, Cancel, Replace, and Insert are shown. The control command advances the pending line after the edits.



2-33. CHARACTER EDIT SUMMARY

A summary of the ways in which characters can be replaced, inserted, and deleted from a line of text is provided in Table 2-3.

Table 2-3. Character Edits Compared

MODE	REPLACE	CANCEL	INSERT
entered by	default R	C	I or S
delimiter action	save current character	delete character	insert blank
tab action	spaces over saving characters	deletes to next stop	inserts blanks to next stop
other characters	replace characters in same relative position	deletes characters	inserts characters

2-34. PATTERN EDITS

Pattern edits involve multiple lines of text from the source work area. There are two kinds of pattern edits: searches and exchanges.

A search matches a character string called a "find field" with a corresponding string in the source work area. There are four search commands (B, D, F, and S) that differ by where they begin the search and the length of the data block which they search.

Exchanges consist of two sets of character strings entered at the keyboard. Whenever the first string is encountered in the source work area, it is replaced by the second string. The commands used to perform exchanges are: G, U, V, X, Y, and Z.

2-35. SEARCH COMMANDS

The B command starts searching at the beginning of the source work area. The D and F commands start searching at the pending line and search to a match or EOF. B implies that the source work area is replaced by the destination work area. The "find field" is only used as a key, and is not actually manipulated as in the exchanges. The S command searches from the beginning of the named file.

"Find field" for search commands may consist of text, or of text preceded by the current delimiter, or an embedded escape character. The "find field" consists of either a character string or NUL. The editor will search for a line containing a matching field in the specified position(s). Only the number of characters specified are used in the match. If the "find field" is omitted, the last "find field" entered is used.

The special characters that may be used in "find fields" are listed below with an explanation of use:

;

The tab control character (; is the default) tabs the "find field" according to the established tab stops (columns 7 and 21 are the default stops). Tabbed-over characters are replaced by blanks in the "find field". For example:

```
LABEL LDA B,I
      LDA B,I
```

are instructions located in the source work area. The following search command will find all instructions containing blanks in columns 1-6 and LDA beginning in column 7.

```
/F;LDA
```

This command will find the second instruction but will not find the instruction beginning with LABEL.

esc or @

The escape character indicates a field of indefinite length and may be used within a "find field". For example:

/F esc XXX esc YYY

This command tells EDITM to find a line containing "XXX" followed by "YYY". "XXX" may be anywhere in the line and "YYY" is constrained only to follow the "XXX" but otherwise may be anywhere in the line.

Note that on some terminals the escape key generates an action within the terminal that is not desirable. In this case, use the alternate escape character (@) set up by the $ command.

/

Delimiter character (/ is the default), is used as the first character of the "find field" and indicates that the search for the match is limited to patterns beginning within the window established by the W command.

CTRL/@

Control @ as the "find field" will find a zero length line.

2-36. B COMMAND — FIND A LINE WITH A FIND FIELD-SOF TO EOF. The B command searches the source work area for the first line of text which matches the "find field" beginning from the start-of-file. All lines passed over in the search are written to the destination work area; they are not deleted. When a line containing the correct character string is found, it becomes the new pending line. The search ends at EOF if no match is found. The B command always forces the destination work area to replace the source work area before the search.

Format:

B CTRL/@	Search for zero length record
B*text*	Search for left-justified text
B (*esc*) ***text***	Search for text anywhere in the line*
B/*text*	Search for text in window*
B;*text*	Search for field at tab stop*
B	Search for field specified in the previous search command

Example:

Find first left-justified HRS.

```
/BHRS
  HRS    BSS 1
```

Find BSS anywhere in line.

```
   (esc)
/B$SS
  HRS    BSS 1
```

Find BSS starting in column 7.

```
/W7
/B/BSS
  HRS    BSS 1
```

Find last specified field.

```
/B
  HRS    BSS 1
/B
  HRS    BSS 1
```

Find first DEF at tab stop.

```
/B;DEF
         DEF HRS
```

*/ in the command format B/text is representative of the initial delimiter. If the delimiter has been changed, the appropriate delimiter character must then be used in place of the /. If the initial tab character (;) is changed, the new tab character must be used. If the alternate escape character is used or is changed, the new character must be used.

2-37. F COMMAND — FIND A LINE WITH A FIND FIELD FROM PENDING LINE TO EOF. This command causes a search of the source work area for a line containing a specified character string. The search begins at the pending line and continues until a match or EOF is encountered. When a line containing this field is found, it becomes the new pending line. If the line is not found, the search ends at the end-of-file.

Format:

F*text*	Search for left-justified field
F (esc) ***text***	Search for field embedded anywhere in a line
F/*text*	Search for field within a window
F;*text*	Search for field beginning at tab stop
F	Search for field specified in the previous search command

Example:

Search for left-justified field DEC. (Not found).

```
/FDEC
  EOF
```

Search for left-justified field HRS.

```
/FHRS
  HRS    BSS 1
```

Search for DEC embedded anywhere in the line.

```
 ┌─esc
 ↓
/FDEC
  TCODE DEC 11
```

The F command followed, directly by text, searches for left-justified occurrences of the field. F, followed by the ESCAPE key and then text, causes the field to be recognized anywhere in the line. If the requested field consists of CTRL/@, the EDITM searches for a zero length line.

Successive searches for the same character string do not require parameters. Once a pattern is entered for this command, it remains in effect until a new pattern is specified. For example:

```
/W7
/F/BSS
  HRS    BSS 1
/F
  MIN    BSS 1
/F
  SEC    BSS 1
```

2-38. D COMMAND — DELETE LINES TO FIND FIELD OR EOF. The D command deletes a block of text from the pending line to the line containing a specified field. After the deletion, the line of text containing the specified field becomes the new pending line. If the specified field is not encountered, the remainder of the file is deleted. If this command is used without specifying a field, the last field entered is used.

Format:

D CTRL/@	Zero length field
D*text*	Left-justified field
D (esc) ***text***	Delete to field anywhere in line
D/*text*	Delete to field in window
D;*text*	Delete to field beginning at tab stop
D	Delete to field specified in previous search command

Example:

Delete from first line to first left-justified occurrence of TCODE field. Display new pending line and verify that this is also the first line of text.

```
/1
  ASMB,R,L,T
/DTCODE
  TCODE DEC 11
/1
  TCODE DEC 11
```

Delete the first occurrence of MIN field anywhere in a line.

```
/1
  ASMB,R,L,T
(esc)
/DMIN
  MIN   BSS <
/1
  MIN   BSS 1
```

Delete to a field within a window.

```
/W11,13
/D/MIN
          STA MIN,I
/1
          STA MIN,I
```

Delete to a field beginning at the first tab stop.

```
/1
  ASMB,R,L,T
/D;STA
        STA HRS,I
```

Delete to the second occurrence of the find field.

```
    /1
      ASMB,R,L,T
esc
/DMIN
  MIN   BSS 1
/D
        STA MIN,I
/1
        STA MIN,I
```

2-39. S COMMAND — MERGE SEGMENT FOLLOWING PENDING LINE. This command allows you to merge a block to text from the source work area to the destination work area or from a specified file to the destination work area. A search for the first find field is started from SOF. Once the first find field is encountered, the second find field is searched for and all text between the first find field and the second find field is merged into the destination work area. If the second find field is not encountered, the rest of the file is merged. The line following the pending line before the S command becomes the pending line upon completion.

This command may be used to correct an accidental delete by specifying the source work area as the merge file and merging the deleted lines. If the file name is not specified, the source work area is assumed.

Format:

Sname,find field 1-find field 2
Slu,find field 1-find field 2

Where:

name
is *filename*[*:security code*[*:cartridge reference number*]]. Refer to File Definition for parameter description. If the source file has been created with a read protect *security code*, this must be specified; *cartridge reference number* should be specified if there is a possibility of another file of the same name on another cartridge; otherwise, *filename* is sufficient.

lu
defines the logical unit number of the input device.

find field
may be any of the following:

text	find left-justified text
/text	find text in window
(esc) *text*	find text anywhere in line
null	find last pattern entered in a search command

Example:

Merge file TIME starting at left-justified field START and ending with left-justified field END after the pending line.

```
/P
  ASMB,R,L,T
/STIME::-2,START-END
```

2-40. EXCHANGE COMMANDS

All of the exchange commands use find fields to locate where the exchange will be made. The character string in the find fields of the G, Y, X, and Z commands function as literals; each character is matched with a character in the source work area. The characters in the find fields of the U and V commands are used as place holders. The find field is used to specify the number of characters to be replaced. The general form is:

G
Y
X
Z } ***old string/new string***

U
V } ***character position/new string***

Remember that the find fields are separated by the delimiter character. If this character is changed, the new one must be used.

2-41. G COMMAND — CHARACTER REPLACE ON PENDING LINE. The G command performs an immediate exchange of an existing character string with a new character string. It performs the exchange on the pending line, displays the result, and the pending line remains the pending line.

The old string may be any length except zero, and the new string any length including zero. Exchanges are made only if the first character of the old string is within the window established by the W command. An exchange pattern remains in effect until a new one is entered.

Format:

Gold string/new string	Replace old string pattern in the pending line with new string
G	Replace using the pattern entered in the previous exchange command

Example:

Set window to include columns 7, 8, and 9. Exchange the occurrence of BSS in the pending line with CRG.

```
/W7,9
/P
  HRS    BSS 1
/GBSS/CRG
  HRS    CRG 1
```

The exchange fields used in the G command are not tabbed, so that the tab character can be used like any other character in the exchange string.

The only time the current delimiter is recognized as a delimiter is the first time it occurs separating the old string and the new string. Any subsequent delimiters are treated as part of the "new" character string. If the delimiter between strings is omitted, EDITM returns the error message "DELIMITER MISSING".

2-42. U COMMAND — UNCONDITIONAL CHARACTER REPLACE (NO LIST). This command is used to insert, delete, or replace the number of characters in the old string with a new string. The characters in the old string are replaced by the new string. The exchange field starts at the first position of the window, but is not limited by its length. The format of this command determines its function.

Format:

U/*new string* ***range command***	Insert new string beginning at window
U*old string*/ ***range command***	Delete old string beginning at window
U*old string*/*new string* ***range command***	Replace old string with new string
U ***range command***	Unconditional character replace using the previous exchange command

Where:

range command
enables the exchange by moving the pending line pointer through the source work area.

Example:

The example below takes a file formatted to store information on disc and reformats it for storage on magnetic tape.

```
/L9
  :ST,8,#LOADR : 17738::7:-1,BA
  :ST,8,#TASMB : 17738::7:-1,BA
  :ST,8,#FTNC1 : 17738::7:-1,BA
  :ST,8,#FTNC2 : 17738::7:-1,BA
  :ST,8,#RTSGN : 17738::7:-1,BA
  :ST,8,#ELQUI : 17738::7:-1,BA
  :ST,8,#RTBTG : 17738::7:-1,BA
  :ST,8,#LABSYS: 17738::7:-1,BA
  :ST,8,LABPRI : 17738::7:-1,BA
  :ST,8,LABMEM : 17738::7:-1,BA
/1
  :ST,8,#LOADR : 17738::7:-1,BA
/U12345/:DU
/99
  EOF
/1
  :DU,#LOADR : 17738::7:-1,BA
```

```
/L9
  :DU,#LOADR : 17738::7:-1,BA
  :DU,#TASMB : 17738::7:-1,BA
  :DU,#FTNC1 : 17738::7:-1,BA
  :DU,#FTNC2 : 17738::7:-1,BA
  :DU,#RTSGN : 17738::7:-1,BA
  :DU,#ELQUI : 17738::7:-1,BA
  :DU,#RTBTG : 17738::7:-1,BA
  :DU,#LABSYS: 17738::7:-1,BA
  :DU,LABPRI : 17738::7:-1,BA
  :DU,LABMEM : 17738::7:-1,BA
/1
  :DU,#LOADR : 17738::7:-1,BA
/W19
/U1234567890/,8
/99
  EOF
/1
  :DU,#LOADR : 17738,8
/L9
  :DU,#LOADR : 17738,8
  :DU,#TASMB : 17738,8
  :DU,#FTNC1 : 17738,8
  :DU,#FTNC2 : 17738,8
  :DU,#RTSGN : 17738,8
  :DU,#ELQUI : 17738,8
  :DU,#RTBTG : 17738,8
  :DU,#LABSYS: 17738,8
  :DU,LABMEM : 17738,8
```

2-43. V COMMAND — UNCONDITIONAL CHARACTER REPLACE (LIST). The V command functions exactly like the U command, but the results of the exchanges are displayed.

Format:

V*/new string* ***range command***	Insert new string beginning at window
V*old string/* ***range command***	Delete old string beginning at window
V*old string/new string* ***range command***	Replace old string with new string
V ***range command***	Unconditional character replace using the previous exchange command

The characters in the old string are replaced beginning at the first column of the window. The old string is not used as a pattern for a search, but rather, to specify the number of characters to be replaced at the start of the window by the new character string. Lines in which the exchange is made are displayed.

Example:

Change the first character of every line in a file to a File Manager command to list (:LI,):

```
/W1
/V1/:LI,
/999
  :LI,CAROL  :52:13
  :LI,TIME   :99:2
  :LI,COMENT:17:2
  EOF
```

2-44. X COMMAND — ENABLE EXCHANGE PATTERN ALL LINES (LIST). This command can be used to make exchanges over a variety of ranges. It is always used in combination with a second command that defines the range; for example, F or M. The X command performs an exchange of an old character string for a new character string. The old string can be any length except zero. The new string can be any length including zero. The exchange takes place when the next command is executed. All lines where an exchange takes place are displayed. The print device is usually the session console, but may be some other device if the second command is L, n, /, or + which specifies an lu parameter. This lu is then used as the display device. The exchange takes place only if the first character of the old string is within the window established by the W command.

The number of lines in which exchanges occur depends on which range command is specified, but generally includes all lines passed over by a search or by positioning to a new pending line. The exception is when the B command is used, and the destination work area replaces the source work area. The exchanges take place between the pending line and EOF. The exchange is then disabled for the search from the beginning of the source work area.

The exchange takes place from the pending line to the EOF. In all cases, the exchange is cancelled after completion of the second command. To do a second exchange with the same pattern, the exchange must be re-enabled using X.

Format:

X*old string/new string* ***range command***	Exchange all occurrences of old string with new string
X ***range command***	Re-enable previous exchange pattern

Example:

Exchange all occurrences of BSS in columns 7, 8, and 9 with CRG.

```
/W7,9
/XBSS/CRG
/999
  HRS   CRG 1
  MIN   CRG 1
  SEC   CRG 1
  TCODE CRG 5
  EOF
```

2-45. Y COMMAND — EXCHANGE ON PENDING LINE, DISPLAY NEXT OCCURRENCE OF PATTERN. The Y command performs an immediate exchange between an existing character string and a new character string. The exchange is always made on the pending line. However, after the exchange the next occurrence of the old string automatically becomes the new pending line, and eligible for an exchange. The old string may be any length, except zero, and the new string may be any length, including zero. Exchanges are made only if the first character of the first string is within the window established by the W command.

Format:

Y*old string/new string*	Exchange old string for new string on pending line
Y	Perform the exchange specified by the previous exchange command

Example:

Exchange the occurrence of BSS in columns 7, 8, and 9 in the pending line with CRG and display next occurrence.

```
/W7,9
/P
  HRS   BSS 1
/YBSS/CRG
  HRS   CRG 1
  MIN   BSS 1
```

After the first exchange, successive exchanges are made by entering Y with no parameters. Once entered, the exchange pattern remains in effect until a new one replaces it. For example:

```
/Y
  MIN   CRG 1
  SEC   BSS 1
/Y
  SEC   CRG 1
  TBUFF BSS 1
/Y
  TBUFF CRG 1
  EOF
```

A combination of Y and F commands can be used to do selective exchanges. The F command is entered (without parameters) to skip an exchange on this occurrence and find the next occurrence. For example:

```
/P
  HRS   BSS 1
/YBSS/CRG
  HRS   CRG 1
  MIN   BSS 1
/F
  SEC   BSS 1
/Y
  SEC   CRG 1
  TBUFF BSS 5
/F
  EOF
```

Since the exchange fields used in the Y command are not tabbed, the tab character can be used like any other character in the exchange string.

Only the first occurrence of the current delimiter is recognized. It is assumed to separate the old string and the new string. Any subsequent delimiters are treated as part of the "new" character string.

2-46. Z COMMAND — ENABLE EXCHANGE PATTERN ALL LINES (NO LIST). The Z command is used for the same purpose as the X command. The only difference between these two commands is that X displays all lines in which exchanges take place; the Z command does not.

Format:

Zold string/new string ***range command***	Exchange all occurrences of old string with new string
Z ***range command***	Re-enable previous exchange pattern

Example:

Exchange all occurrences of BSS with CRG in columns 7, 8, and 9 from pending line to EOF.

```
/W7,9
/ZBSS/CRG
/BBSS
  EOF
```

2-47. TERMINATE COMMANDS

The terminate commands are used to end EDITM operations. With the exception of the ABORT command, they perform any enabled exchanges, cause data remaining in the source work area to go to the destination work area, and cause the edited destination work area to be written to the output file.

All of the normal EDITM termination commands begin with E. Once E is entered, EDITM begins its termination process. Only the second letter of a normal terminate command or rubout may be entered at this time.

Output to a device or a type 0 file can only be done using the ER terminator. This assumes that a directory entry has already been created for type 0 files with the appropriate legal subfunctions for the device are being used.

2-48. A COMMAND — ABORT EDITM

This command is used to end EDITM operations and leave the original source file unchanged. However, the temporary scratch files are purged.

Format:

A

Example:

In this example, EDITM is deliberately aborted by the user.

```
/1
  ASMB,R,L,T
/5
  HRS   BSS 1
/L3
  HRS   BSS 1
  MIN   BSS 1
  SEC   BSS 1
  WEN   NOP
/A
EDITM ABORTED
```

2-49. EC COMMAND — END EDIT AND CREATE A FILE MANAGER FILE

This command ends the edit and creates a named File Manager file for storage of the destination work area.

Format:

EC*name*

:Where:

name
is *filename*[*:security code*[*:cartridge reference number*]]. Refer to File Definition for parameter descriptions.

Example:

In this example, a file is created on-line by EDITM in the destination work area:

```
*ON,EDITM
INPUT?
  EOF
/ ASMB,R,L,T
/       NAM TIME
/       ENT WEN
/       EXT .ENTR,EXEC
/ HRS   BSS 1
/ MIN   BSS 1
/ SEC   BSS 1
/ WEN   NOP
/       JSB .ENTR
/       END
/ECCAROL:LS:-2
END OF EDIT
```

2-50. ER COMMAND — REPLACE OLD FILE WITH NEW FILE

This command ends the edit and replaces an old File Manager file with a new file. You can replace the source file or another file you designate with this command.

Format:

ER	This form can only be used if the source came from a File Manager file originally
ER*name*	Replaces file *name* with the destination work area

Where:

name
is *filename*[*:security code*[*:cartridge reference number*]]. Refer to File Definition for parameter descriptions. If the source file has been created with a write protect *security code*, this must be specified; *cartridge reference number* should be specified if there is a possibility of another file of the same name on another cartridge; otherwise, *filename* is sufficient.

Example:

Use ER to end edit replacing original source with edited text.

```
*ON,EDITM
INPUT? CAROL:LS:-2
  ASMB,R,L,T
/W7,9
/F/END
        END
/I      DEF HRS
/I      JSB EXEC
/ER
END OF EDIT
```

Use ER*name* to end edit replacing original source of File Manager File CRIME with edited text.

```
*ON,EDITM
INPUT? CAROL
  ASMB,R,L,T
/L99
  ASMB,R,L,T
        NAM TIME
        ENT WEN
        EXT .ENTR,EXEC
  HRS   BSS 1
  MIN   BSS 1
  SEC   BSS 1
  WEN   NOP
        JSB .ENTR
        DEF HRS
        JSB EXEC
        END
  EOF
/ERCRIME
END OF EDIT
```

2-51. EN COMMAND — END EDIT AND RENAME DESTINATION WORK AREA

This command ends the edit and allows a scratch file to be renamed as a permanent File Manager file. This command is faster than the EC or ER command, since the file does not have to be copied from the work area.

Format:

EN*name*

Where:

name
is *filename*[*:security code*[*:cartridge reference number*]]. Refer to File Definition for parameter descriptions.

Example:

Use the EN*name* command to rename the destination work area as the File Manager file TIME.

```
*ON,EDITM
INPUT? CAROL
  ASMB,R,L,T
/W7,9
/F/END
        END
/I      DEF *+3
/I      DEF TCODE
/I      DEF TBUFF
/I      LDA TBUFF+3
/I      STA HRS,I
/I      LDA TBUFF+2
/I      STA MIN,I
/I      LDA TBUFF+1
/I      STA SEC,I
/I      JMP WEN,I
/ITCODE DEC 11
/ITBUFF BSS 5
/ENTIME
END OF EDIT
```

EDITM MESSAGES

SECTION III

Errors encountered by the Editor and information messages are reported on the operator console. A summary of the messages, their effects, and error recovery procedures are contained in Table 3-1.

Table 3-1. EDITM Message Summary

MESSAGE	DESCRIPTION	SUGGESTED ACTION
EDITM ABORTED	If this message is displayed after you press "A" to abort EDITM, there is no error. Otherwise, the RTE-M system has aborted EDITM. All source files remain unchanged.	Rerun EDITM.
EOF	A command has caused an attempt to read beyond the current end of the source file.	Return to the beginning of the file by typing in a 0 or 1.
EDITM 0 — INVALID PARAMETER	An illegal parameter was entered.	Retype the command.
EDITM 1 — INVALID COMMAND	An illegal command was entered.	Retype the command. (The "?" command displays all valid commands.)
EDITM 2 — COMMAND FILE NOT FOUND	Error occurred on open of command file.	Rerun EDITM.
EDITM 3 — FILE TOO LARGE	EDITM can only perform this command if there are less than 32K records in the file.	Use alternate command.
EDITM 4 — DELIMITER MISSING	Required delimiter for G, S, U, V, X, Y, or Z command not found.	Retype the command.
EDITM 5 — NO ROOM	Memory buffer space is insufficient to run the Editor.	Load EDITM in a lower area of memory.
EDITM 6 — PARAMETER MISSING	A required parameter has been omitted.	Retype the command.
EDITM 7 — DISC FULL	Edit cannot continue without more disc space or directory space. EDITM terminated.	Pack the disc.
FMP ERROR — xxx	File Manager error number xxx has occurred.	Consult the RTE-M Programmer's Reference Manual.
END OF EDIT	Normal termination.	

SAMPLE EDIT

SECTION IV

This material is provided to show some of the EDITM commands used in combinations. The material is divided into two parts: 1) creation and edit of a comment file, COMMENT, and a FORTRAN program, &FLIST, and 2) a File Manager listing showing the complete program, results of the edits, and the merging of the two files.

EDIT SESSION

```
*ON,EDITM
INPUT?
  EOF
/T;4     ]SET FIRST TAB TO COLUMN 4 TO POSITION COMMENTS.
   T=; D=/ E=@ S=- L=]
/ ;
/ ;
/ ;
/       ]DELETE BELL RING.
/ ;THE PROGRAM FLIST READS A TYPE 3 OR 4 FILE MANAGER FILE
/ ;AND PERFORMS A LIST FUNCTION (SAME AS `:LI,NAMR) TO THE
/ ;LINE PRINTER WITH LINE NUMBERS IN THE FIRST FIVE COLUMNS.
/ ;
/ ;THE SEQUENCE OF OPERATIONS FOLLOWS:
/GFIL/FOL      ]REPLACE CHARACTERS IN LINE JUST CREATED.
     THE SEQUENCE OF OPERATIONS FOLLOWS:
/ ;
/ ;(1)   GET PARAMETERS AND SET UP DEVICES, DEFAULTING
/ ;      IF NECESSARY.
/ ;(2)   GET FILE NAME FROM USER AND OPEN IT.
/ ;(3)   IF FILE ERROR, INFORM USER AND QUIT.
/ ;(4)   IF FILE TYPE NOT 3 OR 4, INFORM USER AND QUIT.
/ ;(5)   INITIALIZE LINE COUNT.
/ ;(5)   READ A LINE FROM THE FILE.
/G5/6     ]CHANGE NUMBER WITH CHAR. REPLACE ON PENDING LINE.
      (6)  READ A LINE FROM THE FILE.
/ ;(7)   IF END OF FILE, WRITE MESSAGE AND QUIT.
/ ;(8)   IF READ ERROR, WRITE MESSAGE AND QUIT.
/ ;(9)   LIST THE LINE ON THE LIST DEVICE.
/ ;(10)  UPDATE THE LINE COUNT AND GO BACK TO STEP (5).
/ECCOMENT
  END OF EDIT

*RU,FMGR
:LI,COMENT
:EX
$END FMGR
```

FILE MANAGER LISTING OF FILE COMENT

```
COMENT T=00004 IS ON CR00099 USING 00004 BLKS R=0000

0001
0002
0003
0004     THE PROGRAM FLIST READS A TYPE 3 OR 4 FILE MANAGER FILE
0005     AND PERFORMS A LIST FUNCTION (SAME AS ':LI,NAMR') TO THE
0006     LINE PRINTER WITH LINE NUMBERS IN THE FIRST FIVE COLUMNS.
0007
0008     THE SEQUENCE OF OPERATIONS FOLLOWS:
0009
0010     (1)    GET PARAMETERS AND SET UP DEVICES, DEFAULTING
0011            IF NECESSARY.
0012     (2)    GET FILE NAME FROM USER AND OPEN IT.
0013     (3)    IF FILE ERROR, INFORM USER AND QUIT.
0014     (4)    IF FILE TYPE NOT 3 OR 4, INFORM USER AND QUIT.
0015     (5)    INITIALIZE LINE COUNT.
0016     (6)    READ A LINE FROM THE FILE.
0017     (7)    IF END OF FILE, WRITE MESSAGE AND QUIT.
0018     (8)    IF READ ERROR, WRITE MESSAGE AND QUIT.
0019     (9)    LIST THE LINE ON THE LIST DEVICE.
0020     (10)   UPDATE THE LINE COUNT AND GO BACK TO STEP (5).
```

EDIT SESSION (continued)

```
*ON,EDITM
INPUT?
  EOF
/ FTN,L,B
/MCOMENT      ]MERGE COMENT FILE INTO NEW FILE BEING CREATED.
  EOF
/1            ]DISPLAY LINE 1 OF SOURCE WORK AREA.
  FTN,L,B
/W1,1       ]SET WINDOW FOR FIRST COLUMN.
/X /C       ]EXCHANGE EVERY BLANK COLUMN 1 FOR A "C".
/FXX        ]FIND A NONEXISTENT PATTERN.
  C
  C
  C
  C  THE PROGRAM READS A TYPE 3 OR 4 FILE MANAGER FILE
  C  AND PERFORMS A LIST FUNCTION (SAME AS ':LI,NAMR') TO THE
  C  LINE PRINTER WITH LINE NUMBERS IN THE FIRST FIVE COLUMNS.
  C
  C  THE SEQUENCE OF OPERATIONS FOLLOWS:
  C
  C  (1)    GET PARAMETERS AND SET UP DEVICES, DEFAULTING
  C         IF NECESSARY.
  C  (2)    GET FILE NAME FROM USER AND OPEN IT.
  C  (3)    IF FILE ERROR, INFORM USER AND QUIT.
  C  (4)    IF FILE TYPE NOT 3 OR 4, INFORM USER AND QUIT.
  C  (5)    INITIALIZE LINE COUNT.
  C  (6)    READ A LINE FROM THE FILE.
  C  (7)    IF END OF FILE, WRITE MESSAGE AND QUIT.
  C  (8)    IF READ ERROR, WRITE MESSAGE AND QUIT.
  C  (9)    LIST THE LINE ON THE LIST DEVICE.
  C  (10)   UPDATE THE LINE COUNT AND GO BACK TO STEP (5).
  EOF
/ C      ]INSERT TEXT AFTER THE PENDING LINE.
/ C    THE PROGRAM STARTS HERE.
/ C
/ ;PROGRAM FLIST(3,90)
/ C
/ ;INTEGER DCB(144),LUN(5),IBUF(40),FNAME(3)
/ C
```

EDIT SESSION (continued)

```
/ ;CALL RMPAR(LUN)
/ ;LUTTY=1
/ ;LULST=6
/ ;IF(LUN(1))2,5,2
/     2 LUTTY=LUN(1)
/     5 IF LUN(2))7,10,7
/W1,150     ]RESET WINDOWS TO MAXIMUM VALUE.
/GIF /IF(     ]G COMMAND USED TO INSERT "(".
      5 IF(LUN(2))7,10,7
/     7 LULST=LUN(2)
/    10 FNAME(2)=2H
/P     ]REQUEST PENDING LINE DISPLAY.
     10 FNAME(2)=2H
/G(2/(1

/G(2/(1     ]G USED TO RESET FNAME VALUE FROM 2 TO 1.
     10 FNAME(1)=2H
/D       //////2    ]D USED TO DUPLICATE FNAME LINE AND CHANGE VALUE.
         FNAME(2)=2H
/D   ]DUPLICATE FNAME(2) LINE.
       FNAME(2)=2H
/G(2/(3   ]REPLACE CHARACTERS "(2" WITH "(3".
       FNAME(3)=2H
/ ;WRITE(LUTTY,100)
/ ;READ(LUTTY,101)FNAME
/ ;CALL OPEN(DCB,IERR,FNAME,0)
/    50 IF(IERR-3)55,20,55
/I;IF(IERR)45,50     ]INSERT TEXT BEFORE PENDING LINE.
/I   45 WRITE(LUTTY,102)FNAME,IERR
/    55 IF(IERR-4)60,20,60
/    60 WRITE(LUTTY,106)FNAME
/ ;GO TO 999
/    20 LINE=1
/    30 CALL READF(DCB,IERR,INBUF,40,LEN)
/ ;IF(LEN+1)35,70,35
/    35 IF(IERR)37,40,37
/    37 WRITE(LUTTY,103)FNAME,IERR
/ ;GO TO 999
/ ;WRITE(LULST,104)LINE,(INBUF(K),K=1,LEN)
/ ;LINE=LINE+1
/ ;GO TO 30
/    70 WRITE(LULST,105)
/ C     ]INSERT BLANK COMMENT LINE.
/ ;F\
   100 FORMAT(//"  ENTER FILE NAME: ")
/   100 FORMAT(3A52)
/G100/101
    101 FORMAT(3A52)
/   102 FORMAT("FILE  `"3A2"' ERROR:""I7/)
/   103 FORMAT("FILE  `"3A2"' ERROR:"I7/)/
   104 FORMAT(X,I4,X,40A2
/   105 FORMAT(//" END-OF-FILE"/"1")
/   106 FORMAT("FILE `"3A2"' OF WRONG TYPE")
/ C
/   999 END
/ ;END$
/P
        END$
/EC&FLIST     ]STORE COMBINED SOURCE IN FILE MANAGER CREATED FILE.
  END OF EDIT

*RU,FMGR
:LI,&FLIST
:EX
$END FMGR
```

FILE MANAGER LISTING OF FILE &FLIST

```
&FLIST T=00004 IS ON CR00099 USING 00008 BLKS R=0000

0001   FTN,L,B
0002   C
0003   C
0004   C
0005   C  THE PROGRAM FLIST READS A TYPE 3 OR 4 FILE MANAGER FILE
0006   C  AND PERFORMS A LIST FUNCTION (SAME AS ':LI,NAMR') TO THE
0007   C  LINE PRINTER WITH LINE NUMBERS IN THE FIRST FIVE COLUMNS.
0008   C
0009   C  THE SEQUENCE OF OPERATIONS FOLLOWS:
0010   C
0011   C  (1)   GET PARAMETERS AND SET UP DEVICES, DEFAULTING
0012   C        IF NECESSARY.
0013   C  (2)   GET FILE NAME FROM USER AND OPEN IT.
0014   C  (3)   IF FILE ERROR, INFORM USER AND QUIT.
0015   C  (4)   IF FILE TYPE NOT 3 OR 4, INFORM USER AND QUIT.
0016   C  (5)   INITIALIZE LINE COUNT.
0017   C  (6)   READ A LINE FROM THE FILE.
0018   C  (7)   IF END OF FILE, WRITE MESSAGE AND QUIT.
0019   C  (8)   IF READ ERROR, WRITE MESSAGE AND QUIT.
0020   C  (9)   LIST THE LINE ON THE LIST DEVICE.
0021   C  (10)  UPDATE THE LINE COUNT AND GO BACK TO STEP (5).
0022   C
0023   C  THE PROGRAM STARTS HERE.
0024   C
0025          PROGRAM FLIST(3,90)
0026   C
0027          INTEGER DCB(144),LUN(5),INBUF(40),FNAME(3)
0028   C
0029          CALL RMPAR(LUN)
0030          LUTTY=1
0031          LULST=6
0032          IF(LUN(1))2,5,2
0033        2 LUTTY=LUN(1)
0034        5 IF(LUN(2))7,10,7
0035        7 LULST=LUN(2)
0036       10 FNAME(1)=2H
0037          FNAME(2)=2H
0038          FNAME(3)=2H
0039          WRITE(LUTTY,100)
0040          READ(LUTTY,101)FNAME
0041          CALL OPEN(DCB,IERR,FNAME,0)
0042          IF(IERR)45,50
0043       45 WRITE(LUTTY,102)FNAME,IERR
0044       50 IF(IERR-3)55,20,55
0045       55 IF(IERR-4)60,20,60
0046       60 WRITE(LUTTY,106)FNAME
0047          GO TO 999
0048       20 LINE=1
0049       30 CALL READF(DCB,IERR,INBUF,40,LEN)
0050          IF(LEN+1)35,70,35
```

FILE MANAGER LISTING OF FILE &FLIST (continued).

```
0051      35 IF(IERR)37,40,37
0052      37 WRITE(LUTTY,103)FNAME,IERR
0053         GO TO 999
0054         WRITE(LULST,104)LINE,(INBUF(K),K=1,LEN)
0055         LINE=LINE+1
0056         GO TO 30
0057      70 WRITE(LULST,105)
0058   C
0059     100 FORMAT(//"  ENTER FILE NAME: ")
0060     101 FORMAT(3AS2)
0061     102 FORMAT("FILE  `"3A2"' ERROR:""I7/)
0062     104 FORMAT(X,I4,X,40A2
0063     105 FORMAT(//" END-OF-FILE"/"1")
0064     106 FORMAT("FILE `"3A2"' OF WRONG TYPE")
0065   C
0066     999 END
0067         END$
```

EDIT SESSION (continued)

```
*ON,EDITM
INPUT?  &FLIST
  FTN,L,B
/43
     45 WRITE(LUTTY,102)FNAME,IERR
/ ;GO TO 999      ]INSERT TEXT AFTER PENDING LINE.
/P
         GO TO 999
/54
         WRITE(LULST,104)LINE,(INBUF(K),K=1,LEN)
/P   40
     40 WRITE(LULST,104)LINE,(INBUF(K),K=1,LEN)
/60
    101 FORMAT(3AS2)
/GAS/A      ]EXCHANGE "AS" TO "A".
    101 FORMAT(3A2)
//
    102 FORMAT("FILE `"3A2"' ERROR:""I7/)
/   103 FORMAT("FILE `"3A2"' READ ERROR:""I7/)
//
    104 FORMAT(X,I4,X,40A2
/GA2/A2)
    104 FORMAT(X,I4,X,40A2)
/ER
  END OF EDIT

*RU,FMGR
:LI,&FLIST
:EX
$END FMGR
```

FILE MANAGER LISTING OF FILE &FLIST

```
&FLIST T=00004 IS ON CR00099 USING 00008 BLKS R=0000
0001   FTN,L,B
0002   C
0003   C
0004   C
0005   C  THE PROGRAM FLIST READS A TYPE 3 OR 4 FILE MANAGER FILE
0006   C  AND PERFORMS A LIST FUNCTION (SAME AS ':LI,NAMR') TO THE
0007   C  LINE PRINTER WITH LINE NUMBERS IN THE FIRST FIVE COLUMNS.
0008   C
0009   C  THE SEQUENCE OF OPERATIONS FOLLOWS:
0010   C
0011   C  (1)   GET PARAMETERS AND SET UP DEVICES, DEFAULTING
0012   C         IF NECESSARY.
0013   C  (2)   GET FILE NAME FROM USER AND OPEN IT.
0014   C  (3)   IF FILE ERROR, INFORM USER AND QUIT.
0015   C  (4)   IF FILE TYPE NOT 3 OR 4, INFORM USER AND QUIT.
0016   C  (5)   INITIALIZE LINE COUNT.
0017   C  (6)   READ A LINE FROM THE FILE.
0018   C  (7)   IF END OF FILE, WRITE MESSAGE AND QUIT.
0019   C  (8)   IF READ ERROR, WRITE MESSAGE AND QUIT.
0020   C  (9)   LIST THE LINE ON THE LIST DEVICE.
0021   C  (10)  UPDATE THE LINE COUNT AND GO BACK TO STEP (5).
0022   C
0023   C  THE PROGRAM STARTS HERE.
0024   C
0025   C     PROGRAM FLIST(3,90)
0026   C
0027          INTEGER DCB(144),LUN(5),INBUF(40),FNAME(3)
0028   C
0029          CALL RMPAR(LUN)
0030          LUTTY=1
0031          LULST=6
0032          IF(LUN(1))2,5,2
0033        2 LUTTY=LUN(1)
0034        5 IF(LUN(2))7,10,7
0035        7 LULST=LUN(2)
0036       10 FNAME(1)=2H
0037          FNAME(2)=2H
0038          FNAME(3)=2H
0039          WRITE(LUTTY,100)
0040          READ(LUTTY,101)FNAME
0041          CALL OPEN(DCB,IERR,FNAME,0)
0042          IF(IERR)45,50
0043       45 WRITE(LUTTY,102)FNAME,IERR
0044          GO TO 999
0045       50 IF(IERR-3)55,20,55
0046       55 IF(IERR-4)60,20,60
0047       60 WRITE(LUTTY,106)FNAME
0048          GO TO 999
0049       20 LINE=1
0050       30 CALL READF(DCB,IERR,INBUF,40,LEN)
```

FILE MANAGER LISTING OF FILE &FLIST (continued)

```
0051          IF(LEN+1)35,70,35
0052       35 IF(IERR)37,40,37
0053       37 WRITE(LUTTY,103)FNAME,IERR
0054          GO TO 999
0055       40 WRITE(LULST,104)LINE,(INBUF(K),K=1,LEN)
0056          LINE=LINE+1
0057          GO TO 30
0058  C
0059     100 FORMAT(//" ENTER FILE NAME: ")
0060     101 FORMAT(3A2)
0061     102 FORMAT("FILE  `"3A2"' ERROR:""I7/)
0062     103 FORMAT("FILE  `"3A2"' READ ERROR:""I7/)
0063     104 FORMAT(X,I4,X,40A2)
0064     105 FORMAT(//" END-OF-FILE"/"1")
0065     106 FORMAT("FILE `"3A2"' OF WRONG TYPE")
0066  C
0067     999 END
0068          END$
```

EDITM IN BATCH ENVIRONMENT

SECTION V

In batch mode, commands to EDITM are supplied as part of a job command file; that is, all EDITM commands must be supplied before EDITM is scheduled. Once EDITM is under way, you can not interact with the computer to change the course of the processing.

All EDITM commands function the same in batch as they do in interactive mode. However, you need not supply the EDITM prompt (/) on the input commands.

It is important that you know beforehand what is going to result from the edit commands in the job command file. Any condition that causes an error message causes EDITM to abort, since there is no possibility of user intervention to fix the error. You must also ensure that EDITM goes through the normal terminate sequence initiated by EC, ER, or EN. If an edit deflects this sequence, EDITM aborts.

When EDITM is executed in batch mode, the EDITM commands must be read from either an external device or a File Manager file. For example, the command file shown in Figure 5-1 performs the same functions as the second editing pass on file &FLIST illustrated in Section IV. If this file was stored in the File Manager file named COMAND, the RTE-M command to schedule EDITM with its input from this file would be:

```
*ON
*RU,EDITM,CO,MA,ND
```

However, if this same command file was stored on cards and the card reader was specified by logical unit 5, the command to schedule EDITM would be:

```
*ON
*RU,EDITM,5
```

```
&FLIST
43
 ;GO TO 999
54
P   40
60
GAS/A
/
   103 FORMAT("FILE `"3A2"' ERROR:""I7/)
/
GA2/A2)
ER
```

Figure 5-1. EDITM Command File

INDEX

In this index the page number for the principal reference of each item is printed in **boldface** type. Page numbers for all other references to the item are printed in standard typeface.

A

A command, 2-3, **2-42**
abort EDITM, 1-5, 2-3, **2-42**

B

B command, 2-2, **2-30**
backup in destination work area, 2-2, **2-19**
batch mode, 1-1, **5-1**
bell control, 2-2, **2-6**
blank command, 2-2, **2-22**

C

C command, 2-2, **2-24**
cancel characters, 2-2, **2-27**
cartridge reference, **1-4**
change special characters, 2-2, **2-4**
character edit combinations, **2-28**
character edit commands, 2-2, **2-24**
character edit summary, **2-28**
character replace on pending line, 2-3, **2-35**
command, A, 2-3, **2-42**
command, B, 2-2, **2-30**
command, blank, 2-2, **2-22**
command, C, 2-2, **2-24**
command, CTRL/C, 2-2, **2-27**
command, CTRL/G, 2-2, **2-6**
command, CTRL/I, 2-2, **2-25**
command, CTRL/R, 2-2, **2-25**
command, CTRL/S, 2-2, **2-26**
command, CTRL/T, 2-2, **2-27**
command, CTRL/@, 2-2, **2-29**
command, D, 2-2, **2-32**
command, escape, 2-2, **2-29**
command, EC, 2-3, **2-43**
command, EN, 2-3, **2-46**
command, ER, 2-3, **2-44**
command, F, 2-2, **2-31**
command, G, 2-3, **2-35**
command, hyphen, 2-2, **2-23**
command, I, 2-2, **2-21**
command, K, 2-2, **2-11**
command, L, 2-2, **2-15**
command, M, 2-2, **2-12**
command, minus, 2-2, **2-23**
command, N, 2-2, **2-18**
command, n, 2-2, **2-16**
command, O, 2-2, **2-24**
command, P, 2-2, **2-14,** 2-24
command, R, 2-2, **2-20**
command, S, 2-2, **2-34**
command, space, 2-2, **2-22**
command, T, 2-2, **2-7**
command, U, 2-3, **2-36**
command, V, 2-3, **2-38**
command, W, 2-2, **2-8**
command, X, 2-3, **2-39**
command, Y, 2-3, **2-40**
command, Z, 2-3, **2-41**
command, esc , 2-2, **2-29**
command, Δ, 2-2 **2-23**
command, !, 2-2, **2-18**
command, #, 2-2, **2-9**
command, $, 2-2, **2-4**
command, +, 2-2, **2-17**
command, −, 2-2, **2-23**
command, /, 2-2, **2-17**, 2-29
command, ;, 2-2, **2-29**
command, =, 2-2, **2-10**
command, ?, 2-2, **2-5**
command, @, 2-2, **2-29**
command,], 2-2, **2-13**
command, ↑, 2-2, **2-19**
command, ^, 2-2, **2-19**
command summary, **2-2**
command syntax, **1-3**
commands, character edits, 2-2, **2-24**
commands, control 2-2, **2-4**
commands, exchange, 2-2, 2-8, **2-35**
commands, line edits, 2-2, **2-20**
commands, pattern edits, 2-2, **2-28**
commands, search, 2-2, 2-8, **2-29**
comments, **2-13**
control commands, 2-2, **2-4**
convention, display format, **1-6**
conventions, keyboard, **1-3**
create file manager file, 2-3, **2-43**
CTRL/D, **1-5**
CTRL/@ command, 2-2, **2-29**
CTRL/C command, 2-2, **2-27**
CTRL/G command, 2-2, **2-6**
CTRL/I command, 2-2, **2-26**
CTRL/R command, 2-2, **2-25**
CTRL/S command, 2-2, **2-26**
CTRL/T command, 2-2, **2-27**
current line, 1-1, **1-2**

In this index the page number for the principal reference of each item is printed in **boldface** type. Page numbers for all other references to the item are printed in standard typeface.

D

D command, 2-2, **2-32**
definition, file, **1-4**
delete a number of lines, 2-2, **2-23**
delete lines to find field or EOF, 2-2, **2-32**
delimiter character, **2-4**
destination work area, **1-1,** 1-2
display a number of lines, 2-2, **2-15**
display a specified line, 2-2, **2-16**
display backup limit, 2-2, **2-18**
display command, 2-2, **2-5**
display commands, 2-2, **2-14**
display error explanation, 2-2, **2-5**
display field, **2-8**
display format, **1-6**
display names of all files, 2-2, **2-18**
display number of characters in pending line, 2-2, **2-18**
display number of words in destination work area, 2-2, **2-18**
display pending line, 2-2, **2-14**
display pending line number, 2-2, **2-18**
display special characters, 2-2, **2-4**
documentation map, RTE-M, **iv**

E

EC command, 2-3, **2-43**
edit pending line, 2-2, **2-24**
edit pending line and advance to next line, 2-2, **2-24**
editing pass, **1-1**
EDITM command summary, **2-2**
EDITM run command, **1-5,** 5-1
EDITM termination, **1-6**
EDITM work areas, **1-1**
EN command, 2-3, **2-46**
enable exchange pattern all lines (list), 2-3, **2-39**
enable exchange pattern for next edit (no list), 2-3, **2-41**
ER command, 2-3, **2-44**
error messages, **3-1**
exchange commands, 2-3, **2-8**
exchange on pending line and display next occurrence, 2-3, **2-40**

F

F command, 2-2, **2-31**
file definition, **1-4**
file manager parameter NAMR, **1-4**
file name, **1-4**
find a line with find field from pending line to EOF, 2-2, **2-31**
find a line with find field - SOF to EOF, 2-2, **2-30**
find a zero length line with find field, 2-2, **2-29**
find field of indefinite length, 2-2, **2-29**
find field tabbed, 2-2, **2-29**
find field within window, 2-2, **2-29**

G

G command, 2-3, **2-35**

H

hyphen command, 2-2, **2-23**

I

I command, 2-2, **2-21**
indefinite find field character, **2-4**
insert characters, 2-2, **2-26**
insert text after pending line, 2-2, **2-22**
insert text before pending line, 2-2, **2-21**
interactive mode, 1-1, **1-5**

K

K command, 2-2, **2-11**
keyboard conventions, **1-3**
kill trailing blanks, 2-2, **2-11**

L

L command, 2-2, **2-15**
limit, memory buffering, **1-2**
line, current, **1-1**
line, pending, **1-2**
line edit commands, 2-2, **2-20**
line numbers, **1-2**
line terminator, 2-2, **2-13**
line terminator character, 2-4, **2-13**
logical rollover, **1-2**

M

M command, 2-2, **2-12**
make a copy of pending line and edit, 2-2, **2-24**
maximum line length for common devices, **2-10**
merge a segment, 2-2, **2-34**
merge source file, 2-2, **2-12**
memory buffering, **1-1**
memory buffering limits, **1-2**
minus command, 2-2, **2-23**
mode, batch, 1-1, **5-1**
mode, interactive, **1-1,** 1-5

N

N command, 1-2, 2-2, **2-18**
n command, 2-2, **2-16**
NAMR parameter format, **1-4**
NAMR file manager parameter, **1-4**

In this index the page number for the principal reference of each item is printed in **boldface** type. Page numbers for all other references to the item are printed in standard typeface.

O

O command, 2-2, **2-24**
output file, **1-1**

P

P command, 2-2, **2-14,** 2-24
pattern edit commands, 2-2, **2-28**
pending line, **1-2,** 1-5
pending line pointer, **1-2**
physical rollover, **1-2**
program listing, **2-1**
prompt character, **1-5**

R

R command, 2-2, **2-20**
relationship between files and work areas, **1-1**
replace characters, 2-2, **2-25**
replace destination scratch file, 2-3, **2-46**
replace old file with new file, 2-3, **2-44**
replace pending line, 2-2, **2-20**
rollover, logical, **1-2**
rollover, physical, **1-2**
RTE-M documentation map, **iv**
running EDITM, **1-5,** 5-1

S

S command, 2-2, **2-34**
S command delimiter, **2-4**
s parameter, **1-5**
sample edit, **4-1**
scratch files, **1-1,** 1-5
search commands, 2-2, 2-8, **2-29**
security code, **1-4**
set line length, 2-2, **2-10**
set tab character and stops, 2-2, **2-7**
set window, 2-2, **2-8**
sequence numbers, 2-2, **2-9**
source file, **1-1,** 1-5
source work area, **1-1,** 1-2
space command, 2-2, **2-22**
space down a number of lines and display, 2-2, **2-17**
syntax, command **1-3**

T

T command, 2-2, **2-7**
tab character, 2-4, **2-7**
tab stops, **2-7**
temporary work areas, **1-1**
terminate commands, 2-3, **2-42**
text lines, **1-1**
truncate remainder of pending line, 2-2, **2-27**

U

U command, 2-3, **2-36**
unconditional exchange (list), 2-3, **2-38**
unconditional exchange (no list), 2-3, **2-36**

V

V command, 2-3, **2-38**

W

W command, 2-2, **2-8**
window, **2-8**
work area, destination, **1-1,** 1-2
work area, source, **1-1,** 1-2
work areas, **1-1**
work areas, temporary, **1-1**

X

X command, 2-3, **2-39**

Y

Y command, 2-3, **2-40**

Z

Z command, 2-3, **2-41**
zero length line, **2-21**

(esc) command, 2-2, **2-29**
Δ command, 2-2, **2-22**
! command, 2-2, **2-18**
# command, 2-2, **2-9**
$ command, 2-2, **2-4**
*ON,EDITM, **1-5,** 5-1
*RU,EDITM, **1-5,** 5-1
+ command, 2-2, **2-17**
− command, 2-2, **2-23**
/ command 2-2, **2-17**, 2-29
; command, 2-2, 2-29
= command, 2-2, **2-10**
? command, 2-2, **2-5**
@ command, 2-2, **2-29**
] command, 2-2, **2-13**
↑ command, 2-2, **2-19**
∧ command, 2-2, **2-19**

Part No. 92064-90004
Printed in U.S.A. 1/77

HEWLETT hp PACKARD

Sales and service from 172 offices in 65 countries.
11000 Wolfe Road, Cupertino, California 95014