

```

*OPT-A
*OUT-R: )
*
*IN-R: ,T: ,S:SRC,D7:SRC1 )
*
*
*↑↑↑↑
*OPT-

```

(TYPE CTRL/P after each file)

Copy the system file PIP from disk to DECTape using filename PIPX.

```

*OPT-S
*OUT-D3:PIPX )
*
*IN-S:PIP )
*↑
*OPT-

```

(TYPE CTRL/P)

3.1.3 Using PIP in an RF08 System

With an RF08 disk as system device, inputs and outputs are specified by S0:,S1:,etc.

Examples:

```

*OPT-S
*OUT-S1:LLDR )
*
*IN-S0:LLDR )
*

```

3.2 EDITOR

Editor (Disk System Editor) enables the user to generate and edit symbolic programs on-line from the Teletype keyboard. The symbolic program may be either printed on the Teletype, punched on paper tape using the high- or low-speed punch, or stored on the system device as an ASCII file.

Editor operates either in command or text mode. In command mode, each typed input is interpreted as a command to perform a certain operation. In text mode, all typed input is interpreted as text to replace, to be inserted into, or to be appended to the contents of the text buffer.

The command language of the Disk System Editor is identical to that of the PDP-8 Symbolic Editor (Chapter 5 of Introduction to Programming) with the following exceptions.

1. CTRL Commands:

CTRL/P During output, progress stops and control returns to command mode.

CTRL/C Always returns control to Monitor.

2. Commands:

P Output entire contents of the buffer followed by a form feed and return to command mode.

nP Output line n, followed by a form feed, return to command mode.

m,nP Output lines m through n, followed by a form feed, return to command mode.

F Illegal command

E Process entire file (perform enough NEXT commands to transfer the remaining input to the output file) and create an end-of-file indicator (legal only for output to the system device).

Certain keys have special operating functions. These keys and their associated functions are listed in Table 3-1.

Table 3-1
Special Key Functions

Key	Mode	Functions
(RETURN)	Text Command	Enter the line in the text buffer. Execute the command.
RUBOUT	Text Command	Delete from right to left one character for each rubout typed (is not in effect during a READ command). Delete entire command.
FORM FEED	Text Command	End of input, return to command mode. List the next line.
CTRL/U	Text Command	Deletes the entire line of text currently being entered and performs a carriage return/line feed. Line counter is unchanged. Cancels commands just typed and performs a carriage return/line feed. * is displayed.
. (period)	Command	Current line counter used as argument alone or in combination with + or - and a number.

Table 3-1 (Cont.)
Special Key Functions

Key	Mode	Functions
/ (slash)	Command	Value equal to number of last line in buffer and used as argument.
LINE FEED	Text	Used in SEARCH command to insert a carriage return/line feed combination into the line being searched.
	Command	List the next line.
ESCape	Command	List the next line.
< (left angle bracket)	Command	List the previous line.
> (right angle bracket)	Command	List the next line.
= (equal sign)	Command	Used in conjunction with . and / to obtain their value (.=0027).
:(colon)	Command	Lower case character, same function as =.
CTRL/TAB	Text	On output, is interpreted as a tab/rubout combination.

Table 3-2 is a summary of Editor commands.

Table 3-2
Summary of Editor Commands

Command	Format(s)	Meaning
READ	R)	Read incoming text and append to buffer until a form feed is encountered. If the low-speed reader is used, Editor pauses when a form feed is encountered so the reader can be turned off. Type any character to return to command mode.
APPEND	A)	Append incoming text from Teletype to any already in the buffer until a form feed is encountered.
LIST	L) nL) m,nL)	List the entire buffer. List the line n. List lines m through n.

(Continued on next page)

Table 3-2 (Cont.)
Summary of Editor Commands

Command	Format(s)	Meaning
OUTPUT (PUNCH)	P)	Output the entire contents of the buffer followed by a form feed and return to command mode. When input is from the low-speed reader and output is to the low-speed punch, a P command causes Editor to pause, to allow time to turn on the punch. Any character may be typed to begin punching. When the specified amount has been punched, the Editor pauses so the punch can be turned off. Type any character to return to command mode. Using a P and then an E, without typing a K in between, causes the buffer to be written twice.
	nP)	Output line n, followed by a form feed.
	m,nP)	Output lines m through n, followed by a form feed.
TRAILER	T)	Punch four inches of trailer.
NEXT	N)	Punch the entire buffer and a form feed; kill the buffer and read next page. When the input is from the low-speed reader, Editor pauses at the end of an N command so that the reader may be turned off. Type any character to return to command mode with text in the buffer. If the tape runs out before the N command is completed, Editor prints a ? and returns to command mode with an empty buffer. The ? indicates that Editor tried to read past the physical end of the input tape. When output is to the low-speed punch, an N command causes Editor to pause before the first punch operation only. If the N command is completed before the entire tape is processed, the Editor pauses after the specified number of N's have been processed. At this point, both the reader and the punch should be shut off. Type any character to return to command mode.
	nN)	Repeat the above sequence n times.
KILL	K)	Kill the buffer.

Table 3-2 (Cont.)
Summary of Editor Commands

Command	Format(s)	Meaning
DELETE	nD)	Delete line n. Editor does not use the core locations made available by line deletion until an N or K command is given.
	m,nD)	Delete lines m through n.
INSERT	I)	Insert before line 1 all text until a form feed is encountered.
	nI)	Insert before line n until a form feed is encountered.
CHANGE	nC)	Delete line n and replace it with any number of lines from the keyboard until a form feed is encountered.
	m,nC)	Delete lines m through n, replace from keyboard as above until form feed is encountered.
MOVE	m,n\$KM)	Move and insert lines m through n before line k.
GET	G)	Get and list the next line beginning with a tag.
	nG)	Get and list the next line which begins with a tag. Start search with line n.
SEARCH	S)	Search the entire buffer for the character specified (but not echoed) after the carriage return; allow modification when found.
	nS)	Search line n, as above, allow modification.
	m,nS)	Search lines m through n, allow modification.
END FILE	E)	Process the entire file (perform enough NEXT commands to pass the remaining input to the output file) then output two form-feed characters signifying an end-of file; legal only for output to the system device. If the low-speed reader is used, the E command causes Editor to pause until the low-speed reader is turned on. After the tape is read and processed the Editor closes the file and returns to Monitor control. If there is no additional tape to be read, typing any character causes the file to be closed normally.

NOTE

If the low speed reader runs out of tape, or if the reader is turned off before a form feed is read, the Editor prints a ? and returns to command mode. The buffer contains material already read. Text entries should be terminated with the FORM FEED character. The text buffer has capacity for 50 to 60 lines of text or 3,600 characters; and the source should be limited to this number of lines or characters. This ensures that no buffer overflow will occur.

If a tape is prepared off-line, care must be taken with respect to form feeds. If input is from the low-speed reader, the Editor ignores the next character after the form feed. For this reason, AT LEAST one blank space should be punched after a form feed in an off-line tape. (Using Editor prepared tapes eliminates this problem.)

Editor prints an error message consisting of a question mark whenever nonexistent information is requested or an inconsistent or incorrect format is used to type a command. The question mark is followed by a carriage return/line feed and the command is ignored.

Editor performs a failsafe operation to preclude the loss of data when the output of text to the disk exceeds the space available. The failsafe technique causes the message FULL to be printed to indicate the buffer is filled. The failsafe operation continues as follows:

1. The E and N commands are disabled. A ? is printed when either is attempted.
2. A P command causes Editor to attempt to save the text by punching it on the high-speed punch. If there is no high-speed punch and a P command is issued the Editor loops, waiting for the high-speed punch flag. At this point the program should be stopped and restarted at location 0177, putting Editor into command mode with the buffer still intact. The buffer can be recovered by using the L command with the low-speed punch turned on.

NOTE

Before the punched output is reloaded into the buffer, the first and last lines of the buffer should be inspected. If the punch was turned on beforehand, the first line is L.RETURN; and the last line is * because Editor returned to command mode. These lines should be deleted.

3. All standard editing functions are available after the failsafe technique has occurred.

If the computer halts at location 2330, a system error has occurred while reading from the disk. Run disk maintenance tests to determine cause of error.

3.2.1 Loading and Saving

Editor is loaded into core from punched paper tape by the Binary Loader. When Editor is in core, it occupies locations shown in Appendix D.

To load Editor into core, call LOAD, using Monitor, and reply to the system responses as explained at the beginning of this chapter and in Paragraph 2.5.

When in core, Editor may be saved on the system device as a system program by Monitor as indicated in Appendix D.

(See Paragraph 2.6.1 for detailed description of the SAVE format.)

A sample printout for loading and saving Editor is:

```
.LOAD)
*IN-R:)
*
*ST=7600)
↑↑
.SAVE EDIT!0-3377;2600)(See Appendix D.)
:
:
```

3.2.2 Operating Procedures

Editor is transferred from the system device into core by Monitor when

EDIT)

is typed.

Editor is now in core and responds by typing

*OUT-

Type one of the following output devices: (T:) for low-speed reader/punch; (R:) for high-speed reader/punch; (S:name) for output to the system device on a file called NAME immediately after OUT-. (E must be typed to properly close the output file.) If the specified device is not valid, that is, not declared when building Monitor, Editor responds with an error message (see Paragraph 2.9) and returns control to Monitor. Refer to Appendix F for a table of valid I/O devices for EDIT.

When Editor recognizes a valid device, it responds with asterisk, carriage return/line feed and *IN-, as shown below.

```
*  
*IN-
```

Specify the input device by typing T: , R: , or S:name or in the same manner as when replying to *OUT-, above.

The Editor responds with

```
*OPT-
```

Specify one of the following options.

- B Preserve blanks. Editor normally replaces multiple blanks (spaces) with tabs, resulting in considerable saving of space on the system device.
- D Enter dynamic deletion mode if input is from the system device, thus allowing space for output if desired. (File name remains on the directory but without any assigned blocks.)
- C Combine the functions of B and D options.
- RETURN None of the above options; assume conversion of two or more blanks to tabs, and not D.

Editor responds to one of the above options with a carriage return/line feed and asterisk. The entire printout might appear as follows.

```
*EDIT)  
*OUT-R: )  
*  
*IN-T: )  
*  
*OPT-B  
*  
*
```

The appearance of the last asterisk in the example above indicates that Editor is ready to accept and operate on the user's symbolic program.

The symbolic program can now be read into core with the R command or typed into core with the A command (see Table 3-2).

Examples:

<u>.LOAD</u>)	Call Loader using Monitor
<u>*IN-R:</u>)	Input to be from high-speed reader
<u>*</u>	Input device valid
<u>*ST=</u>)	Return to Monitor after loading
<u>↑↑</u>	Editor is loaded
<u>.SAVE EDIT!0-3377;2600</u>)	and saved on the system device
<u>.EDIT</u>)	Call Editor using Monitor
<u>*OUT-S:SRC1</u>)	Output to be on system device, file named SRC1
<u>*</u>	
<u>*IN-R:</u>)	Input to be from high-speed reader
<u>*</u>	Input device valid
<u>*OPT-</u>	No blanks, no dynamic deletion mode
<u>*R</u>)	Read incoming text
<u>*E</u>)	Process entire file
<u>.EDIT</u>)	Call Editor using Monitor
<u>*OUT-</u>)	No output, (Don't use dynamic deletion mode or file will be lost.)
<u>*</u>	
<u>*IN-S:SRC1</u>)	Input from filename SRC1
<u>*</u>	File name valid
<u>*OPT-</u>	No option desired
<u>*R</u>)	Read incoming text
<u>*L</u>)	List the entire buffer
<u>*7400</u>	<u>/STARTING ADDRESS OF PROGRAM</u>
<u>ODUM,CLA</u>	
<u>OSR</u>	<u>/GET LOWER LIMIT</u>
<u>DCA LOCK</u>	
<u>HLT</u>	
<u>OSR</u>	<u>/GET UPPER LIMIT</u>
<u>CMA</u>	(CTRL/P was typed here, stopped listing of buffer)
<u>*/L</u>	
<u>\$</u>	
<u>:</u>	(CTRL/C was typed here)

The Disk Editor does not always require specific input or output. For example, when creating a file from the keyboard the following is valid:

```

.EDIT)
*OUT-S:FILE)
*
*IN-)
*
*OPT-)
*A)
TEXT MAY BE KEYED-IN DIRECTLY FROM THE
KEYBOARD.....

```

When lines are deleted the value of the current line counter is not set to the number of the line preceding the deletion. The counter is set to the number of the line deleted, or in the case of multiple deletion, it is set to the number of the line first deleted.

Example:

```

*.L)
      TAD(-1
*. =0024
*.D
*. =0024
*2,5L)
      TAD (-2
      DCA CNT
      TAD TEMP
      JMP .-1

*2,5D)
*. =0002

```

3.3 4K PAL-D DISK ASSEMBLER

The Program Assembly Language for the Disk system (PAL-D), is the symbolic assembly program designed primarily for the 4K PDP-8 family of computers with disk or DECTape. (8K PAL-D is an expanded version of 4K PAL-D and both binary tape and documentation are available from DECUS (Digital Equipment Corporation User's Society) upon request.)

PAL-D is compatible with PAL III except in the way it recognizes Memory Reference Instructions, and with MACRO-8 in respect to the following features: Boolean operators, linkage generation, literals, and a text facility. It does not have user-defined macros, floating point constants, or double precision numbers. The reader should review the 4K ASSEMBLER MANUAL (DEC-08-LAS4A-A-D) before proceeding.

The 4K PAL-D Assembler performs many useful functions, making machine language programming easier, faster, and more efficient. Basically, the Assembler processes the user's source program statements by translating mnemonic operation codes into the binary codes needed in machine instructions, relating symbols to numeric values, assigning absolute core addresses for program instructions and data, and preparing an output listing of the program which includes notification of any errors detected during the assembly process.

Pseudo-operators (pseudo-ops) may be used to direct PAL-D to perform certain tasks or to interpret subsequent coding in a certain manner. Instead of generating instructions or data, pseudo-ops direct the Assembler on how to proceed with the assembly. Pseudo-ops are maintained in the Assembler's permanent symbol table.

The following is a summary of PAL-D's pseudo-ops.