KEYBOARD MONITOR GUIDE

PDP-9

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

PDP-9
ADVANCED
SYSTEM SOFTWARE

KEYBOARD
MONITOR GUIDE

For additional copies of this manual, order No. DEC-9A-NGBA-D from the Digital Equipment Corporation, Program Library, Maynard Mass. 01754 Price $2.00

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

Printed in U.S.A.

HOW TO OBTAIN REVISIONS AND CORRECTIONS

Notification of changes and revisions to currently available Digital software and of new software manuals is available from the DEC Program Library for the PDP-5, 8, 8S, 8I, Linc 8, the PDP-4, 7 and 9 is currently published in DECUSCOPE, the magazine of the Digital Equipment Computer User's Society (DECUS). This information appears in a section of DECUSCOPE called "Digital Small Computer News".

Revised software products and documents are shipped only after the Program Library receives a specific request from a user.

DECUSCOPE is distributed periodically to both DECUS members and to non-members who request it. If you are not now receiving this information, you are urged to return the request form below so that your name will be placed on the mailing list.

To: DECUS Office,
Digital Equipment Corporation,
Maynard, Mass. 01754

☐ Please send DECUS installation membership information.

☐ Please send DECUS individual membership information.

☐ Please add my name to the DECUSCOPE non-member mailing list.

Name _____

Company _____

Address _____

_____

(Zip Code)

TABLE OF CONTENTS

FORTRAN IV	7
FORTRAN IV (abbreviated)	11
MACRO-9	13
MACRO-9 (abbreviated)	17
PIP	21
EDITOR	27
7 TO 9 CONVERTER	33
LINKING LOADER	35
DDT	39
UPDATE	45
DUMP	49
PATCH	51
CHAIN	55
EXECUTE	59
SGEN	61

APPENDIX A
DEVICE ASSIGNMENTS A-1

APPENDIX B
PDP-9 ASCII CHARACTER SET B-1

APPENDIX C
MACRO-9 ERROR DIAGNOSTICS C-1

APPENDIX D
MACRO-9
PERMANENT SYMBOL TABLE D-1

APPENDIX E
EXPLANATION OF IOPS ERROR CODES E-1

APPENDIX F
REVISED FORTRAN IV ERROR LIST F-1

APPENDIX G
FORTRAN IV ERROR LIST G-1

APPENDIX H
FORTRAN IV OTS ERRORS H-1

CONTENTS (cont.)

APPENDIX I
KEYBOARD MONITOR ERRORS I-1

APPENDIX J
SAMPLE PROGRAM USING SYSTEM SOFTWARE J-1

APPENDIX K
EXAMPLE OF BATCH PROCESSING K-1

INTRODUCTION

This guide for operating the bulk-storage version of the PDP-9 ADVANCED Software System is planned for convenient use at the computer. It contains general operating instructions for the Keyboard Monitor, and concise summaries of operating procedures for each individual system program. The user is referred to the following PDP-9 ADVANCED Software System manuals for more detailed descriptions of system programs.

Manual	Document No.
Monitors	DEC-9A-MAB0-D
MACRO-9	DEC-9A-AM9B-D
FORTRAN IV	DEC-9A-AF4B-D
Utility Programs	DEC-9A-GUAB-D

Refer to Appendix I of this manual for a list of Keyboard Monitor errors, and to Appendix J for a sample program using system software.

STARTING THE KEYBOARD MONITOR

Place the system tape on DECtape unit 0 (8). Place the paper tape bootstrap in the paper tape reader and momentarily press the tape feed button to clear the out-of-tape flag. Set the address switches as indicated below, press I/O RESET, and then press READ IN. When the Monitor has been loaded, it types

```
MONITOR
$
```

and waits for a command to be entered on the same line as the dollar sign ($).

The Monitor bootstrap loading addresses are as follows.

17637 for 8K systems
37637 for 16K systems
57637 for 24K systems
77637 for 32K systems

If the bootstrap is already in core, it can be restarted by pressing I/O RESET and START, with the address switches set as follows.

17646 for 8K systems

37646 for 16K systems

57646 for 24K systems

77646 for 32K systems

KEYBOARD MONITOR COMMANDS

Keyboard Monitor commands fall into three categories:

1. Commands that load system programs (terminated with a carriage return (↵) or ALT MODE.
2. Commands to perform special functions.
3. Control character commands, formed by holding down the CTRL key while striking a letter key. These commands are used during the running of system or user programs.

System Program Load Commands

Command	System Program Loaded
F4	FORTRAN IV Compiler
F4A	Abbreviated FORTRAN IV Compiler
MACRO	MACRO-9 Assembler
MACROA	Abbreviated MACRO-9 Assembler
PIP	Peripheral Interchange Program
EDIT	Symbolic Text Editor
CONV	7-to-9 Converter
LOAD	Linking Loader
GLOAD	Linking Loader (set to load and go)
DDT	Dynamic Debugging Technique program
DDTNS	DDT program with no user symbol table
UPDATE	Library File Update program
DUMP	Program to dump saved area (see CTRL Q and QDUMP commands)
PATCH	System tape Patch program
CHAIN	Modified version of Linking Loader -- allows for chaining
EXECUTE (E)	Control program to load and execute chained programs
SGEN	System Generation program

NOTE

The following programs assume that the filename extension is SRC (for source): F4, F4A, MACRO, MACROA, PIP, EDIT, and CONV.

Special Function Commands

Command	Action
LOG (or L)	Can be followed by any comment and terminated by ALT MODE.
SCOM (or S)	Causes typeout of system configuration information, including available device handlers.
API OFF	Disables API.
API ON	Enables API.
QDUMP (or Q)	Conditions Monitor to dump memory on the "save area" of the system tape (or other system device medium if available) in the event of an unrecoverable IOPS error.
HALT (or H)	Conditions the Monitor to halt in the event of an unrecoverable IOPS error.
INSTRUCT (or I)	Types list of Monitor commands.
INSTRUCT (or I) ERRORS	Types system error messages.
REQUEST (or R)	Types .DAT slot assignments and use: a. For system program when followed by system program name. Example: R DDT b. For all positive .DAT slots when followed by USER. Example: R USER c. For all .DAT slots when followed by carriage return. Example: R ↵
ASSIGN (or A)	Allows reassignment of .DAT slots to devices other than those set at system generation time. Example: A PRA -10,3/PPA -6,4

NOTE

.DAT slots used by a system program to be called should in some cases first be checked and modified by use of the REQUEST and ASSIGN commands. Normal .DAT slot assignments are shown in Appendix A. A change of .DAT slot assignments is effective for the current job only since permanent assignments are restored when control is returned to the Monitor. A job is defined as everything that occurs from the time the Monitor takes control and types

```
MONITOR
>
```

until the non-resident portion of the Monitor is called back to core and again types

```
MONITOR
>
```

Command	Action
DIRECT (or D) n	Lists the directory of DECtape mounted on unit n (0-7).
NEWDIR (or N) n	Writes empty directory on DECtape on unit n (units 1-7 only).
GET (or G) n	Restores core image from DECtape (or other system device medium if available) on unit n (0-7).
GET (or G) n address	Restores core image from DECtape (or other system device medium if available) on unit n and restarts at specified address.
GET (or G) n HALT (or H)	Restores core image from DECtape (or other system device medium if available) on unit n and halts.

Control Character Commands

Command	Echos	Action
CTRL S	↑S	Starts user program after loading by linking loader.
CTRL C	↑C	Returns to Monitor; may be used at anytime -- resets all .DAT slot assignments.
CTRL T	↑T	a. Returns control to DDT if DDT is being used. b. Skips to next job when in Batch mode.
CTRL R	↑R	Allows program to continue after IOPS 4 message.
CTRL P	↑P	a. Reinitializes or restarts system program. b. Returns to location specified in user program's last .INIT referencing the Teletype.
CTRL Q n	↑Q	Saves core image on save area of DECtape (or other system device medium if available) mounted on unit n (may be system device) and returns to Monitor.
CTRL U	@	Cancels current line on Teletype (input or output).
RUBOUT	\	Cancels last character input from Teletype (not applicable with DDT).

BATCH PROCESSOR

The Batch Processor portion of the Monitor allows user commands to come from the paper tape reader or card reader instead of the Teletype, thus allowing many programs to be run without operator intervention. All Monitor commands read on the batch device are echoed on the Teletype. Appendix K contains an operational example of batch processing. Monitor commands that are peculiar to the Batch Processor include those listed in the following table.

Command	Function
BATCH (B) dv	Enter Batch mode with dv as batch device; dv can be typed as PR, for paper tape reader, or CD, for card reader
$JOB	Used to separate jobs.
$DATA	Beginning of data -- all inputs up to $END are not echoed on the Teletype.
$END	End of data.
$EXIT	Leave Batch mode.

NOTE

The following commands are illegal when operating in Batch mode: QDUMP, HALT, GET (all forms), BATCH, LOAD, DDT, and DDTNS.

Special Batch Processor control characters include the following:

CTRL T (echos ↑T)	Skip to next job.
CTRL C (echos ↑C)	Leave Batch mode.

To use the Batch Processor, proceed as follows.

a. Load the batch, tape or deck into the batch device.

b. Type BATCH (or B) dv on the keyboard, where dv is PR or CD.

When operating in Batch mode, the Keyboard Monitor has the following operational changes.

a. Any ASSIGN command that references the batch device (any handler) will be assigned to the batch device handler.

b. Any REQUEST command will print the batch device handler as PR* or CD* (whichever applies).

c. When the non-resident Monitor is reloaded, it interprets batch communication bits in the top register of core (177777, 377777, 577777, or 777777):

Bit 0	1 = Batch mode 0 = Non-Batch mode
Bit 1	1 = $JOB command in 0 = Search for $JOB
Bit 2	1 = CD is batch device 0 = PR is batch device

When an error occurs in a job, the non-resident Monitor is reloaded and the Batch Processor skips to the next $JOB command on the batch device.

a. Calling Procedure

The FORTRAN IV compiler is called by typing F4↵ after the Monitor's $ request. When the compiler has been loaded, it types

```
FORTRAN IV
>
```

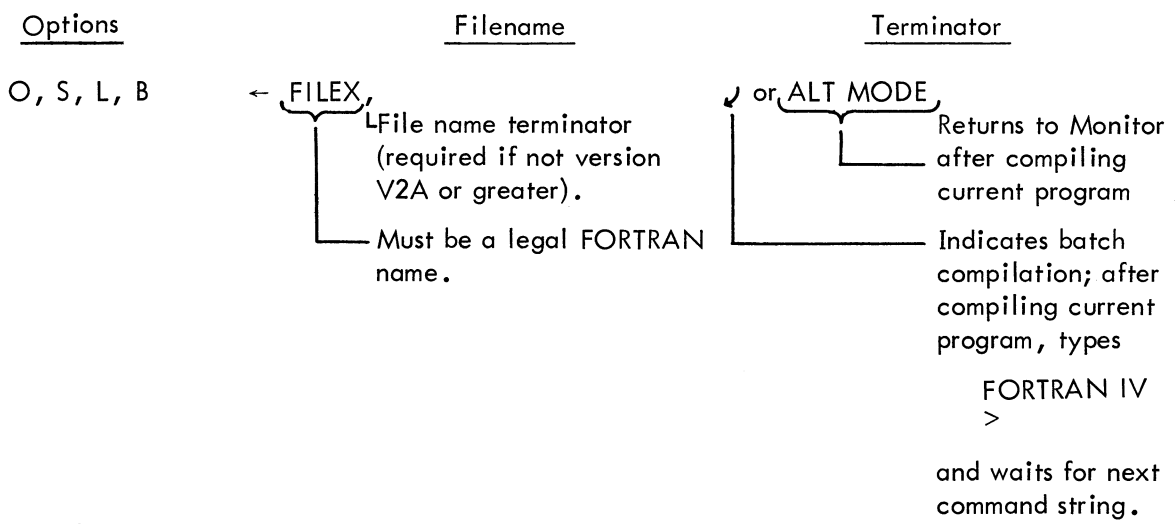
on the Teletype and waits for a command string from the user.

b. General Command Characters

RUBOUT (echos \)	Delete single character.
CTRL U (echos @)	Delete entire line.
CTRL P (echos ↑P)	(1) If paper tape, input, at end of Pass 1, begin Pass 2. (2) While compiler is running, restart at beginning of pass 1.

c. Command String

The format expected by the FORTRAN IV command string processor is as follows.



where

O = Object listing

S = Symbol map

L = Source listing

B = Binary

The options may be used in any combination (or none at all).

The options desired may appear in any order, optionally separated by commas and terminated by ← . If none of the options are wanted, ← is sufficient, with the sole output being compiler diatnostics on the Teletype. Rubouts may be used to delete unwanted characters, and CTRL U (↑U) may be used to delete entire lines prior to typing the command string terminator.

d. Running Instructions

When the compiler is ready,

(1) Place the FORTRAN IV source program on the appropriate input device. (If paper tape, push the tape-feed button to clear the end-of-tape flag).

(2) Type the command string

Paper Tape Input Only*

(3) At the end of Pass 1, the compiler types

END PASS 1
↑P

(4) Replace the source tape in the reader, pushing the tape feed button to clear the end-of-tape flag.

(5) Type CTRL P (↑P) to start Pass 2.

e. Error Conditions and Recovery Procedures

IOPS 4 Device is not ready. Ready device and type

CTRL R (↑R)

IOPS 0-33 Unrecoverable I/O error. Control returns to Monitor (see Appendix E).

See Appendix F for a detailed list of compiler error messages.

f. Restart Procedures

CTRL P (↑P) Restart compiler if running.

CTRL C (↑C) Return to monitor.

g. Examples

(1) To compile a source tape with none of the options, type the command string

← FILEX, ↵

This is very useful for a first compilation when only error messages are desired.

*A paper tape system is assumed; however, if a card reader is available and used, similar procedures apply.

(2) If the output desired is a binary tape, type the command string

B ← FILEX, ↵

(3) If the output desired is a complete listing, type the command string

SLO ← FILEX, ↵

a. Calling Procedure

The abbreviated FORTRAN IV Compiler is called by typing F4A ↵ after the Monitor's $ request. When the compiler has been loaded, it types

```
FORTRAN IV
>
```

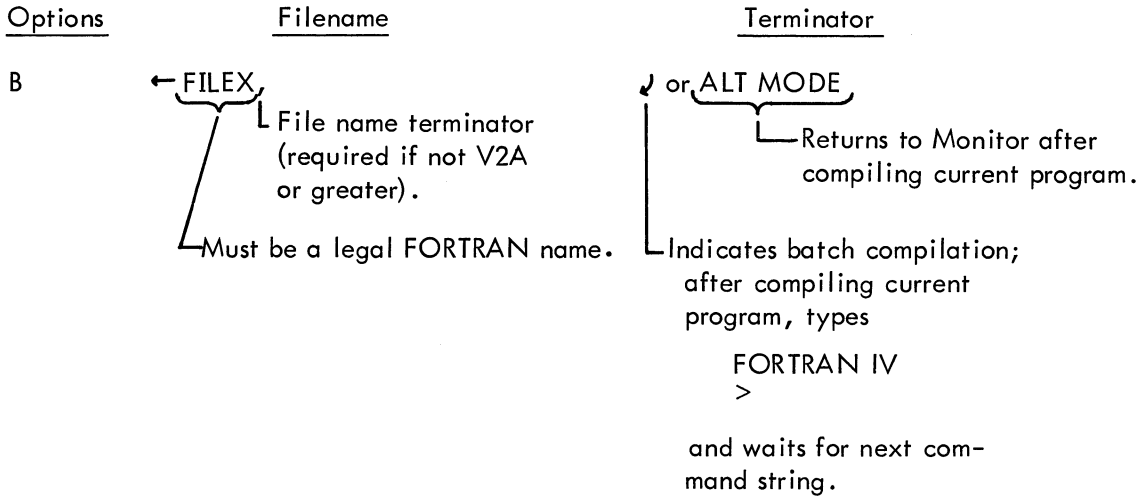
on the Teletype and waits for a command string from the user.

b. General Command Characters

RUBOUT (echos \)	Delete single character.
CTRL U (echos @)	Delete entire line.
CTRL P (echos ↑ P)	(1) If paper tape input, at end of Pass 1, begin Pass 2. (2) While compiler is running, restart at beginning of Pass 1.

c. Command String

The format expected by the FORTRAN IV command string processor is as follows.



where

B = Binary

If the binary output is not wanted, omit the B option (retain the ←) and the only output is compiler diagnostics on the Teletype. Rubouts may be used to delete unwanted characters, and CTRL U (↑U) may be used to delete entire lines prior to typing the command string terminator.

d. Running Instructions

When the compiler is ready,

(1) Place the FORTRAN IV source program on the appropriate input device. (If paper tape, push the tape-feed button to clear the end-of-tape flag.)

(2) Type the command string.

Paper Tape Input Only

(3) At the end of Pass 1, the compiler types:

END PASS 1
↑P

(4) Replace the source tape in the reader, pushing the tape-feed button to clear the end-of-tape flag.

(5) Type CTRL P (↑P) to start Pass 2.

e. Error Conditions and Recovery Procedures

IOPS 4 Device is not ready. Ready device and type

CTRL R (↑R)

IOPS 0-33 Unrecoverable I/O error. Control returns to Monitor. (See Appendix E.)

See Appendix F for a detailed list of compiler error messages.

f. Restart Procedures

CTRL P (↑P) Restart compiler, if running.

CTRL C (↑C) Return to Monitor.

g. Examples

(1) To compile a source tape with no binary output, type the command string

← FILEX,↵

This is very useful for a first compilation when only error messages are desired.

(2) If the binary output is desired, type the command string

B←FILEX,↵

a. Calling Procedure

The MACRO-9 assembler is called by typing MACRO↵ after the Monitor's $ request. When the assembler has been loaded, it types

```
MACRO
>
```

on the Teletype and waits for a command string from the user.

b. General Command Characters

RUBOUT (echos \)	Delete single character.
CTRL U (echos @)	Delete complete line.
CTRL P (echos ↑P)	(1) If paper tape input, at end of Pass 1, begin Pass 2. (2) While assembler is running, restart at beginning of Pass 1.

c. Command String

The format expected by the MACRO-9 command string processor is as follows.

Options	Filename	Terminator
P, S, L, B	←FILEX	↵ or ALT MODE Returns to Monitor after assembling current program Return to MACRO-9 after assembling current program, types MACRO > and waits for next assembly command string.

where

B = Binary

L = Listing

S = Symbol table (on listing device)

P = Parameters to be entered on device assigned to .DAT -10 (must be nonfile oriented)

Options may be used in any combination (or none at all). The options may appear in any order, optionally separated by commas and terminated by ←. If no options are wanted, ← is sufficient and the sole output will be assembly error messages on the Teletype. Rubouts may be used to delete unwanted characters, and CTRL U (↑U) may be used to delete entire lines prior to typing the command string terminator.

d. Running Instructions

When the assembler is ready,

(1) Place the MACRO-9 source program on the appropriate input device. (If paper tape, push the tape-feed button to clear the end-of-tape flag.)

(2) Type the command string.

Paper Tape Input Only

(3) At the end of Pass 1, MACRO types

```
END PASS 1
↑P
```

(4) Replace the source tape in the reader, pushing the tape-feed button to clear the end-of-tape flag.

(5) Type CTRL P (↑P) to start Pass 2.

e. Error Conditions and Recovery Procedures

IOPS 4	Device is not ready. Ready device and type CTRL R (↑R)
IOPS 0-33	Unrecoverable I/O error. Control returns to Monitor. (See Appendix E.)

f. Restart Procedure

CTRL P	Restart the assembler, if running.
CTRL C	Return to Monitor.

g. Examples

(1) To assemble a source tape with none of the options, type the command string:

```
←FILEX↵
```

This is very useful for the first assembly of a program, when only error messages are desired.

(2) If the output desired is a binary tape and input includes parameters to be entered on the secondary input, type the command string

```
P, B ←FILEX↵
```

The parameters should be entered during the first pass only. If the parameters are entered via paper tape, there must be an .EOT on the end of the tape. If the parameters are entered on the Teletype, type CTRL D (EOT) to indicate the end of the parameters. In either case MACRO will type

EOT
↑P

Type CTRL P (↑P) to continue.

(3) If the output desired is a complete listing, but no binary, type the command string

S, L ←FILEX↵

a. Calling Procedure

The abbreviated MACRO-9 assembler is called by typing MACROA↵ after the Monitor's $ request. When the assembler has been loaded, it types

```
MACRO
>
```

on the Teletype and waits for a command string from the user.

b. General Command Characters

RUBOUT (echos \)	Delete single character.
CTRL U (echos @)	Delete complete line.
CTRL P (echos ↑P)	(1) If paper tape input, at end of Pass 1, begin Pass 2. (2) While assembler is running, restart at beginning of Pass 1.

c. Command String

The format expected by the MACRO-9 command string processor is as follows.

Options	File Name	Terminators
P, S, L, B	←FILEX	↵ or ALT MODE Return to Monitor after assembling current program. Return to MACRO-9 after assembling current program, types MACRO > and waits for next assembly command string.

where

B = Binary

L = Listing

S = Symbol table (on listing device)

P = Parameters to be entered on device assigned to .DAT -10 (must be nonfile oriented)

Options may be used in any combination (or none at all). The options may appear in any order, separated by commas and terminated by←. If no options are wanted, ← is sufficient and the sole output will be assembly error messages on the Teletype. Rubouts may be used to delete unwanted characters, and CTRL U (↑U) may be used to delete entire lines prior to typing the command string terminator.

d. Running Instructions

When the assembler is ready,

(1) Place the MACRO-9 source program on the appropriate input device. (If paper tape, push the tape-feed button to clear the end-of-tape flag.)

(2) Type the command string.

Paper Tape Input Only

(3) At the end of Pass 1, MACRO types

```
END PASS 1
↑P
```

(4) Replace the source tape in the reader, pushing the tape-feed button to clear the end-of-tape flag.

(5) Type CTRL P (↑P) to start Pass 2.

e. Error Conditions and Recovery Procedures

IOPS 4 Device is not ready. Ready device and type

CTRL R (↑R)

IOPS 0-33 Unrecoverable I/O error. Control returns to Monitor. (See Appendix E.)

Refer to Appendix C for MACRO-9 Error Diagnostics.

f. Restart Procedures

CTRL P (↑P) Restart assembler, if running.

CTRL C (↑C) Return to Monitor.

g. Examples

(1) To assemble a source tape with none of the options, type the command string

←FILEX↵

This is very useful for the first assembly of a program, when only error messages are desired.

(2) If the output desired is a binary tape and input includes parameters to be entered on the secondary input, type the command string

P, B ←FILEX↵

The parameters should be entered at the start of the first pass only. If the parameters are entered via paper tape, there must be an .EOT on the end of the tape. If the parameters are entered on the Teletype, type CTRL D (EOT) to indicate the end of the parameters. In either case, MACRO will type

```
EOT
↑P
```

Type CTRL P (↑P) to continue.

(3) If the output desired is a complete listing but no binary, type the command string

```
S, L ← FILEX↵
```

a. Calling Procedure

The Peripheral Interchange Program (PIP) is called by typing PIP↵ after the Monitor's $ request. When PIP has been loaded, it types

PIP

on the Teletype and waits for a command string from the user.

b. General Command Characters

RUBOUT	(echos \ )	Delete single character.
CTRL U	(echos @)	Delete entire line.
CTRL P	(echos ↑P)	Restart PIP.

c. Command String

The general format of a PIP command string is as follows.

F DDU:FILEO;EXT(S) ← SDU:FILE1;EXT

It is usually terminated by a carriage return or ALT MODE. The colons and semicolons in the command string may be replaced by spaces.

F is a function character, which may be:

T = Transfer file
V = Verify file
S = Segment file
L = List directory
D = Delete file
C = Copy
R = Rename file
B = Block copy
N = New directory

DDU is the destination device and unit number, if applicable

PP = Paper tape punch
DT = DECtape
TT = Teletype

LP = Line printer

MT = Magnetic tape

DK = Disc

FILEO: EXT is the output file name and extension and may be omitted if the output device is non-file oriented.

(S) indicates the switch options

Data Mode Switches:

A = IOPS ASCII

B = IOPS binary

I = Image alphanumeric

H = Image binary

D = Dump

Function switches:

G = Correct bad parity lines

E = Convert tabs to spaces

C = Convert multiple spaces to tabs

Y = Segment files

W = Combine files

S = Create new system directory

N = New directory

F = Insert form heads

← terminates information concerning the destination device. Data for the source device follows the←.

PR = Paper tape reader

TT = Teletype

CD = Card reader

DT = DECtape

MT = Magnetic tape

DK = Disc

Carriage return or ALT MODE is the command string terminator.

Carriage Return	Return to PIP after completion of the current function.
ALT MODE	Return to Monitor after completion of the current function.

Rubouts may be used to delete unwanted characters, and CTRL U (↑U) may be used to delete the entire line prior to typing the command string terminator.

d. Operating Instructions

The following tables summarize legal switch/operation combinations within a Keyboard Monitor environment.

Legal Operation/Switch Combinations

Operation	Legal Switches
Transfer File (T)	A,B,I,H,D,E,G,C,W,Y,N,S
Verify File (V)	A or B
Segment File (S)	(None)
List Directory (L)	N or S or None
New Directory (N)	(None)
Delete File (D)	(None)
Rename File (R)	(None)
Copy Tape (C)	N or S or H or None
Block Copy (B)	N or S or None

Legal Switch Combinations for Transfer File

Switches	A	B	I	H	D	E	G	C	W	Y	N	S
E	✓						✓		✓	✓	✓	✓
G	✓					✓		✓	✓	✓	✓	✓
C	✓						✓		✓	✓	✓	✓
W	✓	✓				✓	✓	✓			✓	✓
Y	✓					✓	✓	✓			✓	✓
N	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
S	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		

e. Error Conditions and Recovery Procedures

Error Message	Recovery
COMMAND STRING TOO LONG, TRY AGAIN	Retype command string.
ILL. FUNCTION	Retype from function character on.
ILL. DEV. OR UNIT ILL. DEV. OR UNIT TERMINATOR DEV. ILL. FOR OPTION OR FUNCTION AND DIRECTION	Retype from device name on.
DEV. (UNIT) NOT IN + DAT TABLE ILL. SYS. DEV. IN DAT SLOT 1	Type ↑C to restore Monitor and perform ASSIGN
SYS. TAPE NOT ON UNIT 0	Mount System Tape on Unit 0 and retype command string.
TOO MANY FILES OR BLKS., TRY AGAIN	Retype command string.
TOO MANY CHARS. IN FILE OR EXT. NAME SOURCE FILE NOT ON DEV.	Retype from File Name on.
TOO MANY SOURCE FILES TOO MANY DEST. FILES	Check number of files actually transferred and type another command string to transfer remainder.
DATA MODE NEEDED	Type data mode in parentheses followed by carriage return.
SWITCH ILL. FOR DEV. ILL. SWITCH SWITCH CONFLICT SWITCH ILL. FOR FUNCTION	Retype from switch on.
ILL. TERMINATOR	Retype from terminator on.
INPUT PARITY ERR.	If binary, check data. If ASCII, retype command string using G switch.
INPUT CHECKSUM ERR. ASCII INPUT LINE TOO LONG	Check data.
ILL. BLK.#	Retype from block # on.
READ - COMP. ERR. ON BLK. N	When operation complete, try B function on error block.
S OPERATION NOT PERFORMED	Execute S operation; then retype T command.
STRINGS 1 TO 16 ACCEPTED	Perform segmentation; then further segment last destination file.
TOO FEW DEST. FILES FOR # OF SEGMENT POINTS	Retype command string with correct # of destination files. (1 more than # of segmentation points).

f. Restart Procedures

CTRL P Restart PIP.

CTRL C Return to Monitor.

g. Examples

(1) To transfer an ASCII paper tape to the DECtape on Unit 1

T DT1 NAME SRC(A) ← PR↵ or T DT1:NAME;SRC(A) ← PR↵

(2) To list an ASCII file from DECtape on the line printer

T LP (A) ←DT1 NAME SRC or T LP: (A) ← DT1:NAME;SRC↵

(3) To rename a binary file on DECtape

R DT1 NEW BIN ←DT1 OLD BIN or R DT1:NEW; BIN←DT1:OLD;BIN↵

(4) To list the directory of a DECtape on Unit 1

L TT → DT1↵

(5) To transfer a binary file from the DECtape on Unit 1 to that Unit 2, zeroing out the directory on the DECtape on Unit 2

T DT2 NAME BIN (BN) ← DT1 NAME BIN↵ or

T DT2:NAME; BIN (BN) ← DT2:NAME;BIN↵

(6) To copy the entire DECtape on Unit 2 on to the DECtape on Unit 1

C DT1← DT2

(7) To verify an ASCII file on a DECtape on Unit 3

V DT3 FILEA SRC (A)↵ or V DT3:FILEA;SRC (A)↵

(8) To copy four blocks from a DECtape on Unit 4 to one on Unit 7

B DT7 → DT4 5, 15, 165, 1075↵

(9) To generate a new directory (clear the old directory) on the DECtape on Unit 4

N DT4↵

(10) To segment an ASCII tape on the DECtape on Unit 1 into four paper tapes

S TAG1, TAG2, TAG3↵

T PP ,,, (AY) ← DT1 NAME SRC↵

a. Calling Procedure

The Editor is called by typing EDIT↵ after the Monitor's $ request. When the Editor has been loaded, it types

```
EDITOR
>
```

on the Teletype and waits for a command from the user.

b. General Command Characters

RUBOUT	(echos \)	Delete single character.
CTRL U	(echos @)	Delete entire line.
CTRL P	(echos ↑ P)	Restart the editor.

c. Command String

Not applicable.

d. Operating Procedures

Editing Operation 1: Creating a file. (When Editor is brought in core, it awaits an OPEN command if the input device is file oriented.)

	User Types in	Action	Effect
(1)	OPEN filename↵	INPUT	Mode is changed from Edit to Input.
(2)	Content of the program (each line is terminated by↵)	Puts out previous line typed	Line typed in is processed.
(3)	↵ (necessary before CLOSE	EDIT >	Change from Input to Edit Mode.
(4)	CLOSE filename↵	EDITOR >	Closes created file.

Editing Operation 2: Modifying an existing file: Place the input file on the appropriate input device. Open the file with OPEN NAME . The user may then use any of the edit commands summarized below.

SUMMARY OF EDITING COMMANDS

Editor-Monitor Communication

Command	Abbreviation	Activity
EXIT	n/a	Transfer control to Monitor.

File Housekeeping

OPEN nm ext	n/a	Prepare input file (named "nm ext") for editing. (SRC assumed if no ext given.)
CLOSE	n/a	Terminate editing on input file.

Locative Requests

FIND string	F	Bring first line beginning with "string" to work area.
LOCATE string	L	Bring first line containing "string" to work area.
NEXT	N	Bring next consecutive line to work area.
BOTTOM	B	Bring last line of file to work area.
TOP	T	Reset pointer to beginning of file.
PRINT	P	Print the current line on the Teletype.

Manipulative Requests

DELETE	D	Discard the current line.
RETYPE string	R	Replace current line with "string."
INSERT string	I	Add "string" as a complete line to the file after (below) the current line.
CHANGE /string1/string2/	C	Replace, in the current line, the first occurrence of "string 1" with "string 2."

NOTE

The slash delimiters may be replaced by any of the 64 ASCII characters; however, the character should be chosen such that it is not present in "string1" or "string2".

APPEND string	A	Add "string" at the rightmost end of the current line.
VERIFY ON/OFF	V	Set verify mode to print (ON) or ignore printing (OFF) lines after processing CHANGE, LOCATE, and FIND requests.
BLOCK ON/OFF	n/a	Set program to operate in block mode (ON) or in line-by-line mode (OFF).

Manipulative Requests (Cont)

Command	Abbreviation	Activity
BRIEF ON/OFF	n/a	Set brief mode to print truncated (ON) or full (OFF) lines.
OVERLAY	O	Delete current +(n-1) lines, change to input mode and insert after current line.
Input/Output Requests		
OUTPUT ON/OFF	n/a	Output/No output.
READ	n/a	Fill block buffer from input file.
WRITE	n/a	Add block buffer to output file.
GET	G	Add lines from subsidiary input device after (below) current line.
Miscellaneous Requests		
SIZE	S	Set total lines to occupy block buffer.
INSERT	I	Change mode to input.

e. Error Conditions and Recovery Procedures

(1) END OF FILE (MEDIUM) REACHED BY

End of file or buffer has been reached by the indicated command.

(a) Use TOP command to return to first line.

(b) If GET command, continue editing.

(c) With paper tape input or output, use CLOSE and reload tape in reader.

(2)

The indicated command is not legal.

Note that there must be a space between the command and its argument and that the period (.) is not recognized as a current line indicator.

(3) READ ERROR

Parity or checksum error on indicated input device.

(4) TRUNCATED

Indicated line greater than 90 characters.

NOTE

The user has a choice, following either of the above errors, of either modifying the line that caused the error (via any manipulative request) or of allowing the line to stand as is in the output file (via any locative request).

(5) BUFFER CAPACITY EXCEEDED BY

Block-mode buffer overflow caused by indicated line.

(6) FILE filename ext NOT FOUND

The Editor assumes that the user wishes to create a new file with the given name, and changes to input mode.

(7) NO FILE NAME GIVEN

No file name given in either OPEN or CLOSE request. Repeat request giving file name.

(8) IOPS 4

Device is not ready. Ready device and type control R (↑R).

(9) IOPS 0-33

Unrecoverable I/O error. Control returns to Monitor. (See Appendix E.)

f. Restart Procedures

CTRL P (↑P)	Program restart when Editor is waiting for a command.
CTRL C (↑C)	Control mode change, if in input mode. Return to Monitor.

g. Examples

Purpose	Original	Desired Change	Command (user types)
To change 1 character in a word	JMP TAG1	JMS TAG1	C /P/S/
To eliminate 1 character in a word	JMS* LOOP	JMS LOOP	C /*// or C /S*/S/
To add a string of characters at the end of a line	→| DAC CNTR	→| DAC CNTR/ counter check	A /counter check
To print the current line			P ↵
To read the next line			N ↵
To change mode (from edit to input or vice versa)			↵

"CLOSE" should always be the last command issued to complete editing.

How to Use BLOCK MODE:

User types in:

BLOCK ON	Begin BLOCK mode.
SIZE N	N=Number of lines in block (assumes 55 lines if unspecified).
READ	N lines are brought in core.
WRITE	Output all lines.
BLOCK OFF	Return to line-by-line editing.

a. Calling Procedure

The 7-to-9 Converter is called by typing CONV↵ after the Monitor's $ request. When the converter has been loaded, it types

```
7-TO-9 CONVERTER
>
```

on the Teletype and waits for a command string from the user.

b. General Command Characters

RUBOUT	(echos \)	Delete last character in command string; may be repeated n times to delete n characters.
CTRL U	(↑U) (echos @)	Delete entire line.
CTRL P	(↑P)	(1) Reinitialize converter. (2) Resume operation after placing new tape in reader.

c. Command String

The format expected by the Converter command string processor is as follows.

Options	Filename	Terminator
L,A,R,E,Tn	← FILE 1, FILE 2 Output program name and input program name, if different	↵ or ALT MODE Return to Monitor after completion of job. Return to converter to perform more conversions, types 7-TO-9 CONVERTER > and waits for next string.

where

L = Listing

A = Insert .ABS pseudo op

R = Remove origin settings

E = .EOT instead of .END

T = Combine input tapes

n = Decimal number of input tapes

Options may be used in any combination (or none at all). The options desired may appear in any order, separated by commas and terminated by ←. If no options are wanted, ← is sufficient. Rubouts may be used to delete unwanted characters, and CTRL U (↑U) may be used to delete entire lines prior to typing the command string terminator. If an error in the command string is detected, CONV types:

COMMAND STRING ERROR
>

and waits for a new command string.

d. Operating Instructions

The program to be converted must be ready on the appropriate input device before the command string is typed. (If paper tape, push the tape-feed button to clear the end-of-tape flag.) Each input tape is considered as a complete job, unless the Tn option is used. The second file name is needed only if input and output are file-oriented and it is desired to change the name. The output name is placed in the new .TITLE statement and is used as the name of the file. File extension SRC is assumed.

e. Error Conditions and Recovery Procedures

COMMAND STRING ERROR	Retype command string.
IOPS 4	Device is not ready. Ready device and type CTRL R (↑R)
IOPS 0-30	Unrecoverable I/O error. Control returns to Monitor. (See Appendix E.)

f. Restart Procedure

CTRL P (↑P)	Reinitialize converter.
CTRL C (↑C)	Return to Monitor.

g. Examples

(1) To convert a single tape, with .ABS insertion and a listing, the command string would be

L, A ←NAME ↵

(2) To convert and combine four tapes, with no listing, the command string would be

T4 ← NAME ↵

a. Calling Procedure

The Linking Loader is called by typing LOAD↵ or GLOAD↵ after the Monitor's $ request. LOAD is used to load-and-halt; GLOAD is used to load-and-go. When the loader is ready, it types:

```
LOADER
>
```

on the Teletype and waits for a command string from the user.

b. General Command Characters

RUBOUT	Delete last character typed. n rubouts may be used to delete n characters within a program name. (There is no character echo.)
CTRL P (↑P)	(1) Continue loading (paper tape input)
CTRL S (↑S)	(2) Start user's program (if GLOAD not used).

c. Command String

The command string may have several different forms as follows (the >'s are supplied by the loader).

```
>NAME1, NAME2, NAME3 (ALT MODE)
                or

> NAME1↵
> NAME2↵
> NAME3 (ALT MODE)
                or

> ,, (ALT MODE) – Valid for paper tape input only
```

It is important to accurately specify the number of programs (n) to be loaded with n-1 commas or carriage returns before the ALT MODE.

d. Operating Procedure

The program to be loaded must be ready on the appropriate input device before the command string is typed. If the input is on DECtape, any subprograms must be on the same tape as the main program.

The loader types out the name and address of each program, subprogram, and library routine loaded.

If GLOAD was used to call the loader, execution will start automatically. If LOAD was used, the loader will type ↑S, when loading is complete. The user then starts his program by typing CTRL S.

e. Error Messages

.LOAD 1	Memory overflow	Unrecoverable loader errors
.LOAD 2	Input data error	
.LOAD 3	Unsatisfied global symbol (missing program)	
.LOAD 4	Illegal .DAT slot request by user program	
.IOPS 4	Device not ready. Ready device and type CTRLR.	
.IOPS 0-33	Unrecoverable I/O error. Control returns to Monitor. (See Appendix E.)	

f. Restart Procedures

CTRL C (↑C) Return to Monitor.

g. Examples

```
LOADER
>EX1  (ALT MODE)
EX1        17365
↑S                    Type CTRL S to start program.
```

```
LOADER
>(ALT MODE)           Program name not needed with paper tape input to loader.
EX1        17365
↑S
```

```
LOADER
>EX2,SUB (ALT MODE)
EX2        17656
SUB        17613
.DA        17544
BCDIO      14551
STOP       14536
SPMSG      14442
FIOPS      13712
```

```
OTSER    13604
REAL     12651
↑S

LOADER
>EX2                    Carriage return may be used in place of comma.
>SUB (ALT MODE)
EX2      17656
SUB      17613
.DA      17544
BCDIO    14551
STOP     14536
SPMSG    14442
FIOPS    13712
OTSER    13604
REAL     12651
↑S

LOADER
EX2 (ALT MODE)
EX2      17656
BCDIO    14663
STOP     14650
SPMSG    14554
FIOPS    14024
OTSER    13716
REAL     12763
SUBROT   ØØØØØ          The subroutine was omitted.
.LOAD 3                 Unsatisfied global symbol.
```

a. Calling Procedure

The DDT (Dynamic Debugging Technique) program is called by typing DDT↵ or DDTNS↵ after the Monitor's $ request. (The use of DDTNS prevents loading of the user's symbol table, thus saving space.) When DDT has been loaded, it types

```
LOADER
>
```

on the Teletype and waits for a command from the user.

b. General Command Characters

RUBOUT	Delete last character typed (during load phase only).
CTRL P (↑P)	During load phase, continues loading with new tape.
CTRL T (↑T)	Restart DDT or bypass loading.

c. Command String, Loader Phase

The command string may have several different forms as follows (the >'s are supplied by loader portion of DDT).

```
> NAME1, NAME2, NAME3 (ALT MODE)
```

or

```
> NAME1↵
> NAME2↵
> NAME3 (ALT MODE)
```

or

```
>,, (ALT MODE) – Valid for paper tape input only
```

It is important to specify the number of programs (n) to be loaded with n-1 commas or carriage returns before the ALT MODE.

d. Operating Procedures

The program to be loaded must be ready on the appropriate input device before the command string is typed. If the input is on DECtape, any subprograms must be on the same tape as the main program.

The loader types out the name and address of each program, subprogram, and library routine loaded.

When loading has been successfully completed, DDT types

```
DDT
>
```

Debugging may now begin.

Following is a summary of DDT commands. For detailed information on the operation of each command, refer to the DDT section of the Utility Programs Manual (Doc. No. DEC-9A-GUAB-D).

SUMMARY OF COMMANDS

Linkage Characters

+	Arithmetic plus
-	Arithmetic minus
(space)	Field separator

Breakpoints

k n"	Insert breakpoint at location k, assign number n (1-4).
n"	Remove breakpoint number n (1-4).
"	Remove all existing breakpoints.
!	Restart from breakpoint.
n!	Restart from breakpoint, wait n times before reentering breakpoint.
↑T	Interrupt processing, go to DDT-9.

Examinations and Modifications

k/	Open location k.
↵	(Carriage return) – Close the location.
↵	(Line feed) – Close the location, open next location.
↑	(Up arrow) – Close the location, open the preceding location.
↑ Z	(CTRL Z) – Close the location, open addressed location, continue original sequence.
↑ A	(CTRL A) – Close the location, open addressed location, start new sequence.
↑ X	(CTRL X) – Close the location, open the location addressed by 15-bit transfer vector, start new sequence.
NUM$	Type contents as 6-digit octal numbers.
TV$	Type contents as transfer vectors.
SYM$	Type contents as symbolic instructions (assumed by default).
:	Retype in alternate mode (NUM$, SYM$).
=	Retype as transfer vector.
REL$	Type addresses as relative to defined symbols (assumed by default).
RLC$	Type addresses as relocatable numbers.
ABS$	Type addresses as absolute numbers.

Starts and Restarts

'	Start user's program at normal starting point.
k'	Start user's program at location k.
!	Restart user's program from breakpoint.
n'	Restart user's program from breakpoint, waits n times before reentering breakpoint.
↑T	(Control T) – Interrupt processing.

Searching Operations

k EQ$	Search for words equal to k.
k UN$	Search for words not equal to k.
k ADR$	Search for instructions with effective address equal to k.

Special DDT-9 Locations

AC$	Holds AC at a breakpoint.
LNK$	Status of Link at a breakpoint.
MSK$	Contains search mask.
LO$	Lower limit of search.
HI$	Upper limit of search.
PA$	First unused location in patch area.
AX$	Number of auto-index used by breakpoints.
RF$	Current relocation factor.
SA$	Normal starting address.
Bn$	Address of breakpoint n (1-4).

Symbol Definition

s)	Assign symbol s to the current location.
K(s)	Assign symbol s to location k.

Patch File Output

PFO$	Patch file output (from LO$ to HI$, inclusive).
k PFO$	Single location patch file output.
SNS$	Save new symbols.
PFE$	Close patch file output.

Patch File Input

PFI$	Read patch file.

Coresident Subroutines

k HDR$	Use symbol table and relocation factor of subroutine k.
HDR$	Use symbol table and relocation factor of main program.

Miscellaneous Features

Q$	Contents of currently open location.
.	Address of currently open or most recently opened location.
&	Bypass mnemonic instruction lookup.
$k^{\#}$	Execute the instruction k.
↑U	Cancel the line.
↑T	Interrupt processing.

e. Error Conditions

(1) Loader Messages:

.LOAD 1	Memory overflow
.LOAD 2	Input data error
.LOAD 3	Unsatisfied global symbol (missing program)
.LOAD 4	Illegal .DAT slot request by user program

(2) DDT Running Errors:

OVERFLOW	Too many new symbols defined. Current entry ignored.
ERROR	Read error on patch file input. All patches loaded before error are good.
?	General error indication. Current entry ignored. Possible causes are listed below.

Undefined symbol
Address above core
Incorrect command
Illegal character

(3) I/O Errors:

.IOPS 4	Device is not ready. Ready device and type CTRL R (↑R).
.IOPS 0-33	Unrecoverable I/O error. Control returns to Monitor during loading phase and to DDT during debugging phase. (See Appendix E.)

f. Restart Procedure

CTRL T	CTRL T (↑T) Restarts DDT
	CTRL C (↑C) Return to Monitor

g. Examples

```
LOADER
> EX1  (ALT MODE)
EX1          14455
DDT
>
```

```
LOADER
>EX2,SUB (ALT MODE)
EX2          14746
SUB          14703
.DA          14634
BCDIO        11641
STOP         11626
SPMSG        11532
FIOPS        11002
OTSER        10674
REAL         07741
DDT
>
```

```
LOADER
> (ALT MODE)          Program name not needed with paper tape
EX1          14455   input to loader.
DDT
> NUM$
```

```
> AC$/      000000
 LNK$/      000000
 MSK$       777777
 LO$/       014455  =    BEGIN      Low limit of program.
 HI$/       015007  =    END+15     High limit of program.
 PA$/       002420                  Low limit of available memory.
 AX$/       000017
 RF$/       014455                  Relocation factor.
 SA$/       414455  =    BEGIN      Starting address.
 B1$/       000000
 B2$        000000
 B3$        000000
 B4$        000000
> BEGIN/    000776
> SYMS
> ./    CAL+776
 BEGIN+1/   CAL+1
 BEGIN+2/   CAL+14455 = BEGIN
 BEGIN+3/   CAL
 BEGIN+4/   LAC END+1
 READ-5/    JMS TYPE
 READ-4/    LAC END+2
 END+2/     LAW 17774
 READ-3/    DAC COL
>
```

a. Calling Procedure

The Library Update Program is called by typing UPDATE ↵ after the Monitor's $ request. When the Update program has been loaded it types

```
UPDATE
>
```

on the Teletype and waits for a file specifying command from the user.

b. General Command Characters

RUBOUT (echos \)	Delete last character in command string. May be repeated n times to delete n characters.
CTRL U (U) (echos @)	Delete entire line

c. Command String

The user should first type, on the same line as the right angle bracket (>), a file specifying command string in the following format.

Options	File Name	Terminator
L, U, N	← FILEX	↵ or ALT MODE

where

L	= Library file listing on .DAT-12
U	= Update from .DAT-14 to .DAT-15 with secondary input on .DAT-10
N	= Create from .DAT-10 onto .DAT-15
Neither U or N	= Input on .DAT-14 (no output on .DAT-15 or secondary input on .DAT-10) used primarily with CLOSE command to get clean library file listing on .DAT-12.

The default library file name is .LIBR (the file name used in library .SEEK's by the Linking Loader). The file name extension is always assumed to be BIN.

If the file specifying the command string is terminated by

1. ALT MODE, control will be returned to the Monitor when updating of the current file is completed.
2. Carriage return (↵), control will remain with UPDATE when work on the current file is complete and it will output

```
UPDATE
>
```

to the teleprinter to indicate readiness for the next file specifying command.

d. Operating Procedures

When UPDATE is ready for a library file manipulation command, it outputs > to the teleprinter. The user should now type a file manipulation command on the same line as the right angle bracket (>) terminated by carriage return (↵) and in the following format.

DELETE (D) NAME	Delete the named routine from file, copying all previous routines (valid command only if U option).

REPLACE (R) NAME 1, NAME 2	Replace NAME 1 with NAME 2 (default is NAME 1), copying all previous routines (valid command only if U option). Replacement comes from .DAT-10.
INSERT (I) NAME 3, NAME 4	Insert NAME 3 after NAME 4 (default is last routine processed or beginning of file) (copying all previous routines if U option). Specifying a second argument (NAME 4) is only valid in U mode. Insert comes from .DAT-10.
END (E)	Position at end of file (copying all routines if U option).
KILL (K)	Abort operations on current file, destroying bad output file. Issued when user detects trouble with updating process.
CLOSE (C) FILENM	Performs END if not done; clears up at end of update satisfying all options and giving the output file the name FILENM (default is .LIBR even if a NAME was given in file specifying command string, i.e.; U← NAME↵). EXIT to Monitor or remain in UPDATE for next file as a function of the file specifying command string terminator (ALT mode or ↵).

The library file listing on .DAT slot -12 will be in the following format.

LIBRARY FILE LISTING FOR FILENM PAGE 1

PROGRAM NAME	PROGRAM SIZE	ACTION
	0	DELETE NAME
NAME 2	477_8	REPLACE NAME 1, NAME 2
NAME 4	352_8	
NAME 3	517	INSERT NAME 3, NAME 4
⋮	⋮	⋮

e. Error Conditions and Recovery Procedures

Error messages - recoverable

If command completely unintelligible

```
?
>
```

If Delete, Replace, Insert (with 2 arguments) used with other than U option

```
VALID ONLY IN U MODE - COMMAND IGNORED
>
```

If Delete, Replace, Insert (with 1 argument) used without U or N option

```
VALID ONLY IN U OR N MODE - COMMAND IGNORED
>
```

If no name given after Insert, Delete, Replace

```
ILLEGAL COMMAND STRUCTURE - COMMAND IGNORED
>
```

If program requested in any command not found in forward direction (tape at end)

```
EOF REACHED BY SEARCH - COMMAND IGNORED
>
```

This file is still open and may be accessed via INSERT, CLOSE and KILL commands. If wrong program used as input on .DAT slot -10 for Replace or Insert command

```
WRONG PROGRAM AS INPUT - CORRECT INPUT AND ↑P
```

Set up input device with the correct program and then type ↑P on the keyboard.

Error messages - terminal (new file specifying command required)

If end code found before program name on binary input

```
PROGRAM NAME MISSING - DYNAMIC KILL
UPDATE
>
```

If not enough room in core for program

```
BUFFER OVERFLOW - DYNAMIC KILL
UPDATE
>
```

If read error on input buffer

```
UNRECOVERABLE READ ERROR ON .DAT N - DYNAMIC KILL
UPDATE
>
```

f. Restart Procedures

CTRL P	Restart Update program.
CTRL C	Return to Monitor.

g. Example

1. To Update FILEA:

```
UPDATE
>U←FILEA↵          /File specifying command must be first
>I NAME2, NAME3↵   /Insert routine NAME2 after NAME3
>R NAME4, NAME5↵   /Replace routine NAME4 with NAME5
>D NAME1↵          /Delete routine NAME1 from file
>C FILEA↵          /Close FILEA
UPDATE             /Returns to UPDATE since
>                  /file specifying command above
                   /was terminated with a↵
```

2. To update BCDIO on the systems .LIBR file (user responses are underlined)

```
$ A DTA0-14/DTA1-15/PRA-10↵   /Scratch tape on -15
$ UPDATE↵                     /Call Update
UPDATE
>U←↵                          /Specify Update function
> R BCDIO↵                    /Replace BCDIO with new version
>C↵                           /Close the file
UPDATE                        /New .LIBR on -15
```

```
>↑C                                  /Return to Monitor
MONITOR
$PIP↵
PIP
>D DT0 .LIBR BIN↵                   /Delete old library
>T DT0 (B)←DT1 .LIBR BIN↵           /Put new .LIBR on systems tape
>
```

a. Calling Procedure

The Dump program is called by typing DUMP↵ after the Monitor's $ request. When the Dump program has been loaded, it types

```
DUMP
>
```

on the Teletype and waits for a command from the user.

b. General Command Characters

RUBOUT (echos\)	Delete last character in command string. May be repeated n times to delete n characters.
CTRL U (↑U) (echos@)	Delete entire line

c. Command String

The formats expected by the DUMP command string processor are as follows.

Command	Function
ALL	The entire ↑Q area (from location 10 to the address in .SCOM) on the device associated with .DAT slot-14 (at ↑Q time, this device was the specified output device) is listed on the device associated with .DAT slot -12.
XXXXX-YYYYY (XXXXX≥10, and YYYYY≤C(.SCOM)	The ↑Q area between absolute addresses XXXXX and YYYYY on the device associated with .DAT slot -14 is listed on the device associated with .DAT slot -12. At ↑Q time, this device (.DAT slot -14) was the specified output device and XXXXX and YYYYY were the absolute (octal) bounds of the core area to be dumped.
ZZZ#	The content of block #ZZZ on the device associated with .DAT slot -14 is listed on the device associated with .DAT slot -12. The block number is in octal radix.

NOTE

If the Listing output (.DAT slot -12) is to a file oriented device, the file is named MEMORY and has the extension DMP.

d. Operating Procedures

Not applicable.

e. Error Conditions

Any unrecognizable command will cause a question mark (?) to be typed on the Teletype. Control is then returned to the command string processor which types > to indicate its readiness for a command.

f. Restart Procedures

If a command is terminated by a carriage return (↵), control returns to the command string processor after completion of the request.

```
DUMP
>
```

will be printed on the Teletype indicating readiness for another command.

If a command string is terminated by the ALT MODE character, control returns to the Monitor upon completion of the request.

g. Example

To dump locations 16730 through 16750:

```
MONITOR
$ASSIGN DTD0-14↵
$ DUMP↵
DUMP
>16730-16750↵
 16730 000032 003740 013777 000000 000000 413420 013422 463356
 16740 127400 463356 127400 000612 003766 003773 000000 020202
 16750 000000
DUMP
  >
```

a. Calling Procedure

The Patch program is called by typing PATCH ↵ after the Monitor's $ request. When the Patch program is loaded, it types

```
PATCH
>
```

on the Teletype and waits for a command from the user.

b. General Command Characters

RUBOUT (echos\)	Delete last character in command string. May be repeated n times to delete n characters.
CTRL U (↑U) (echos@)	Delete entire line.
CTRL P (↑P) (echos ↑P)	Restart Patch program.

c. Command String

Not applicable.

d. Operating Procedures

Before calling in PATCH, the user should ensure that appropriate device handlers are assigned to .DAT slots as indicated below. This can easily be accomplished by means of the REQUEST and ASSIGN commands to the monitor.

.DAT Slot	Handler	Function
-14	DTA0, or DKA0, or MTA0	Input from and output to the system device
-10	PRA	Input from the paper tape reader

Patch uses .DAT slot -3 for Teletype output and .DAT slot -2 for input from the keyboard (or batch input device). The user cannot modify .DAT slots -3 and -2.

NOTE

Before typing any commands to PATCH, ensure that the mode switch for the system device is on WRITE ENABLE.

The Patch program recognizes four commands as follows.

1. Selecting a system program for patching
2. LIST
3. READ
4. EXIT

System Program Selection Command. - The user must specify a system program to be patched before issuing a LIST or READ command. This is accomplished by typing the system program name after the right angle bracket (>) and terminating with ALT MODE or carriage return. All LIST and READ commands refer to the previously selected system program, until another name is given in a command. DDT, CHAIN, and the Loader cannot be edited with the Patch program since they are relocatable system programs. System program names that are recognized by Patch are as follows.

CONV	
DUMP	MACROA
EDIT	PATCH
EXECUTE	PIP
F4	.SGEN1
F4A	.SGEN2
KM9	.SYSLD
MACRO	UPDATE

The system program selection command has the following form:

> NAME ↵

When the named system program is ready for modification, the Patch program will type a right angle bracket and await another command.

LIST Command. - The LIST command has the following form:

> L OCTADR ↵

As indicated, the user simply types L, followed by a space, followed by an octal address, and terminated with a carriage return or ALT MODE. The octal address must be an address within the range of the currently selected system program. On the next line, the PATCH program then prints either a space or >, followed by the octal address, followed by a /, followed by the contents of that address in octal, and terminated by a > as in the following example.

```
> L 132 ↵
  00132/777435>
```

If the user wishes to modify the contents of the location just printed, he may type the octal value with which the old contents are to be replaced. If he does not want to change the old contents, he must terminate the line with ALT MODE or carriage return without typing any number. Carriage return closes the current location and opens the next higher location. For example

```
> L 256 ↵
  00256/734202>774202 ↵
  00257/600511>
```

Ending the line with ALT MODE closes the current location and terminates the listing sequence.

READ Command. - The READ command has the following form:

> READ ↵

Before issuing a read command, the user must place the paper tape in the reader and momentarily depress the tape-feed button to clear the NO-TAPE-IN-READER flag. Immediately after the > at the beginning of a line, type READ, terminated by ALT MODE or carriage return. The mode switch on the system device should be on WRITE at all times. The paper tape must be in absolute binary block format, optionally headed by the ABS BIN LOADER. The tape may be a completely new version of the system program or it may be patched to certain registers within the program. Both are handled identically. PATCH reads one block at a time from the paper tape. For each data word in the paper tape block, the program calculates the address within the system program. If that address is within the current system device block in core, the new contents (from paper tape) replace the old contents of that block. If the address is in a block not currently in core, the current block is written out and the new one is brought in.

NOTE

Programs on the system device are straight core dumps onto contiguous blocks of 400 (octal) words each. When PATCH is called to LIST and/or modify a location in a program, it checks to see whether the contents of that location are within the current block in core. If not, it checks to see if the current block in core was modified. If so, it writes that block onto the system device before reading in the next block. A space preceding the printout of the octal address/octal contents indicates that that address is in the current core block. If a > is printed instead of a space, it indicates that the address was not in the resident block and that a new block was read in. In this case, all patches preceding the last > at the beginning of a line have been entered in the program on the system device. If error messages occur, the error handler first writes the current block onto the system device if the block has been modified.

EXIT Command. - The EXIT command has the following form:

>EXIT ↵

Control is returned to the Monitor when this command is typed.

e. Error Conditions

If an error is detected when reading from paper tape, PATCH will terminate reading and will print out the cause of the error. If the user makes a format error in his data or command line, PATCH will terminate the current command and print an appropriate error message. All errors handled by PATCH cause the current block in core to be written onto the system device if that block had been modified. The following is a list of error messages:

Message	Cause of Error
ILLEGAL COMMAND	Not a legal system program name or PATCH command; or, first command was list or read with no program selected.
NOT OCTAL DIGIT	The address in a list command or the modification data in a list command sequence, contained a character that was not an octal digit.
TOO MANY DIGITS	The user-typed octal number contained more than six octal digits.
ADDRESS OUT OF RANGE	The address to be listed and/or modified is outside the range of the current system program.
CHECKSUM ERROR	Bad data read in from paper tape.
END OF MEDIUM	End of paper tape, detected. No start block found.

In addition to the above errors, IOPS errors 2 and 4 may occur. If IOPS error 2 occurs, control is returned to the Monitor. The user should assign the proper device (to .DAT slot -14 or -10 as indicated in Paragraph d of this section) and return to PATCH. If IOPS error 4 occurs, the user should ready the appropriate device and type CTRL R.

f. Restart Procedures

CTRL P (↑P)	Restart Patch program
CTRL C (↑C)	Return to Monitor
>EXIT ↵	Return to Monitor

g. Examples

Excluding the comments on the right, the following is a sample listing of what would appear on the Teletype after a session using PATCH. Characters typed by the user have been underlined for clarity.

```
MONITOR
$ PATCH ↵                          /Call in Patch

PATCH V1A
>MACRO ↵                           /Select Macro
>L 100 ↵                           /List location 100
>00100/000000>10 ↵                 /Modify. List 101
 00101/777777>777776 ↵             /Modify. List 102
 00102/000033>↵                    /No change. List 103
 00103/000000>447(ALT MODE)        /Modify. End sequence
>L 476↵                            /List location 476
 00476/600113>↵                    /No change. List 477
 00477/741012>741102 ↵             /Modify. List 500
>00500/600254>↵                    /New block read in
 00501/200636>(ALT MODE)           /No change. End sequence
>READ ↵                           /Read from paper tape
>PIP ↵                            /Change to PIP
>READ ↵                           /Read from paper tape
>EXIT ↵

MONITOR
$
```

a. Calling Procedure

The Chain program is called by typing CHAIN ↵ after the Monitor's $ request. When the Chain program is loaded it types

```
CHAIN
>
```

on the Teletype and waits for a command from the user.

b. General Command Characters

RUBOUT (echos\)	Delete last character in command string. May be repeated n times to delete n characters.
CTRL U (↑U) (echos@)	Delete entire line.

c. Command String

Not applicable.

d. Operating Procedures

CHAIN is a relocatable system program which builds an XCT type file. Input consists of the standard relocatable binary (from F4 or MACRO) with appropriate calls to CHAIN for segment loading. Output is an XCT type file which can be loaded and run with the monitor's EXECUTE command. The following .DAT slots are used by CHAIN.

.DAT-6	Output of XCT file
.DAT-5	External Library
.DAT-4	User Program(s)
.DAT-3	Control and Error Messages
.DAT-2	Command String
.DAT-1	System Library

The Chain program recognizes six commands:

1. BUILD FILENM
2. CHAIN (C) N
3. FILE1, SUB1, etc.
4. END (E)
5. CLOSE
6. EXIT

BUILD FILENM - This command initiates the building of FILENM XCT onto .DAT-6. If no file name is given an error message will occur. This command is legal only immediately after the typeout of CHAIN. If it is used at any other time, an error message will be given and the BUILD command ignored.

CHAIN(C) N - A chain with number N is begun at this point. N may be any decimal number. It must be greater than any N given in a previous CHAIN command. This command is legal only after a BUILD or an END command. If it is used at other times it will be ignored and an error message given.

FILE1, SUB1, etc. - All commands immediately following the CHAIN command and before the END command will be interpreted as filenames to the Linking Loader portion of the XCT file builder. The following are illegal file names: BUILD, CHAIN, C, END, E, and CLOSE.

NOTE

File names may be separated by comma, space, carriage return, or ALT MODE.

END(E) - Terminates the filenames used for a particular chain. This command must be used after a CHAIN command and with at least one filename between it and the CHAIN command. It may appear on the same line as the filenames if so desired.

CLOSE - Finish building the file FILENM XCT, and restart CHAIN.

EXIT - Return to Monitor

e. Error Conditions

?
Illegal command

ILLEGAL DECIMAL DIGIT
Illegal decimal digit in chain number

ILLEGAL COMMAND ORDERING
Command is out of order and should not be used at this point

ILLEGAL CHAIN NUMBER
Chain number is less than or equal to the last chain number

ILLEGAL FILE NAME
File name used is same as reserved command

.LOAD N Errors
See Linking Loader section of Utility Programs Manual (Doc. No. DEC-9A-GUAB-D)

f. Restart Procedures

CLOSE	Finish building FILENM XCT, and restart CHAIN
EXIT	Return to Monitor

g. Examples

```
CHAIN
>BUILD TEST              Initiates building of TEST
>CHAIN 1                 First chain
>FILE1, SUB1, SUB2       Programs in this chain
>SUB3, SUB4
>END                     End of this chain
        .
        .
        .
      Memory allocation typeouts
        .
        .
        .
.
.
.
>CHAIN 4                 Last chain
>FILE4, SUBA
>SUBB, END               End of this chain
        .
        .
        .
      Memory allocation typeouts
        .
        .
        .
>CLOSE                   Terminate TEST
CHAIN
>EXIT                    Return to Monitor
```

Memory Allocation Typeout

1. Load addresses of programs, subroutines, and library routines loaded

FILE	XXXX
SUB	XXXX
LIB1	XXXX
⋮	
LIBN	XXXX

2. Special typeouts

CHAIN#	N	(Chain number)
LOWEST	XXXX	(Lowest register used) (Contents of .SCOM+3 plus 1) (C(.SCOM+3)+1)
COMSZE	XXXX	(Size of blank common)

a. Calling Procedure

The command to EXECUTE is given at load time and consists of the file name of the XCT type file that is to be run. It has the following form.

```
MONITOR
$EXECUTE        FILEN
```

or

```
MONITOR
$E    FILEN
```

b. General Command Characters

Not applicable.

c. Command String

Not applicable.

d. Operating Procedures

Not applicable.

e. Error Conditions

WARNING - COMMON SIZE DIFFERS

Blank common size of a new chain is different from that of the previous chain.

RESTART INPUT & ↑P

An attempt was made to call a chain number less than the current chain using nonbulk storage input.

```
.SEG    01
        no chain can call itself
.SEG    02
        EOF reached without finding requested chain
.SEG    03
        End of Medium reached without finding requested chain
.SEG    04
        Read error on .DAT-4
.SEG    05
        Blank common overlap the requested chain
```

NOTE

All the .SEG type errors are fatal and cause a return to the monitor to be made via an .EXIT command.

Prior to requesting the System Generator via the SGEN Keyboard command, request information about the current operating environment by entering the following device examination and information keyboard commands.

SCOM	Causes output of certain system information, including a list of the device handlers available and a brief description of each of their features.
REQUEST SGEN	Causes output of .DAT slot assignments used by the System Generator and the use made of each .DAT slot. If any assignment does not agree with the user's need, it can be changed via the ASSIGN keyboard command (being aware of the handlers available as listed by the SCOM command and the handler requirements of SGEN).

NOTE

It is imperative that your old system device be assigned to .DAT slots -10 and -14 and the unit that will contain your new system device be assigned to .DAT-15.

When the System Generator is loaded (via the SGEN command) and ready to begin questioning the user on items pertinent to the building of a new system tape, it outputs the following introduction and then proceeds with the questioning.*

```
SYSTEM GENERATOR

THIS PROGRAM WILL GENERATE A NEW SYSTEM TAPE
ON THE DEVICE SPECIFIED IN .DAT SLOT -15.  IT WILL
DETERMINE THE CHARACTERISTICS OF THIS SYSTEM
TAPE BY ASKING YOU A SERIES OF QUESTIONS.
IF IT CANNOT UNDERSTAND THE ANSWER YOU GIVE,
IT WILL REPEAT THE QUESTION.  HERE GOES!

HOW MUCH CORE IS AVAILABLE? TYPE 8, 16, 24, or 32.
>16
IS AN API AVAILABLE?  TYPE Y OR N.
>N
IS AN EAE AVAILABLE?  TYPE Y OR N.
>Y
IS TELETYPE A MODEL 33?  TYPE Y OR N.
>N
INDICATE THE PRESENCE OR ABSENCE OF THE FOLLOWING DEVICE
HANDLERS BY TYPING Y OR N:
```

*The answers to questions must be terminated by a carriage return.

```
PRA?        >Y
PRB?        >Y
PPA?        >Y
PPB?        >N
PPC?        >Y
LPA?        >N
CDE?        >N
CDB?        >N
DTA?        >Y
DTB?        >Y
DTC?        >Y
DTD?        >Y
ARE ANY OTHER DEVICE HANDLERS PRESENT?  TYPE Y OR N.
>Y
HOW MANY?  TYPE OCTAL NUMBER.
>1
TYPE THREE CHARACTER HANDLER NAME FOR NO. 01.
>AAA
HOW MANY SKIP IOTS SHOULD BE IN SKIP CHAIN FOR THIS
DEVICE HANDLER?  TYPE OCTAL NUMBER.
>1
TYPE UP TO FIVE CHARACTER MNEMONIC FOR SKIP IOT NO. 01,
A COMMA, AND OCTAL SKIP IOT.
>ASKIP,701111↵
THE FOLLOWING SKIP IOTS ARE TO BE INCLUDED IN THE
SYSTEM SKIP CHAIN:

CLSF
KSF
TSF
RSF
PSF
DTDF
DTEF
ASKIP

TYPE THEM IN SKIP CHAIN ORDER, ONE PER NUMBER.
(PRECEED SKIP BY A MINUS IF REVERSE SKIP IOT.)
NOTE:  USE ↑P TO RETURN TO THIS POINT.
NO. 01?     >ASKIP
NO. 02?     >DTDF
NO. 03?     >CLSF
NO. 04?     >RSF
NO. 05?     >PSF
NO. 06?     >KSF
NO. 07?     >TSF
NO. 10?     >DTEF
WHICH DEVICE HANDLER IS TO BE USED FOR THE SYSTEM DEVICE?
(NOTE:  THIS MUST BE SMALLEST INPUT ONLY HANDLER)
>DTC
```

NOTE

The example assumes user needs device handler for a special device (e.g., an A/D converter).

NOTE

Structure the skip chain so that those devices requiring faster service have their skip IOT(s) at the beginning of the chain. (Example assumes that the users special handler, AAA requires top priority.)

```
TYPE THE DEVICE HANDLER NAME (NON FOR NONE) AND THE UNIT NO. FOR THE
FOLLOWING .DAT SLOTS:
NOTE: USE ↑P TO RETURN TO THIS POINT.

-15?        >DTA4
-14?        >DTA3
-13?        >DTB1
-12?        >TTA
-11?        >DTB3
-10?        >PRA
 -6?        >PPC
 -5?        >DTC2
 -4?        >DTC1
 -1?        >DTC0
  1?        >TTA
  2?        >TTA
  3?        >DTA1
  4?        >TTA
  5?        >DTA1
  6?        >PRA
  7?        >AAA
 10?        >NON

THANKS FOR THE INFO.  THIS WILL TAKE A FEW MINUTES.
```

NOTE

Refer to Appendix A for system program .DAT slot requirements in setting up the negative .DAT slots. The positive .DAT slots belong to the user, and the standard setup should be a function of the needs of the users at your installation.

NOTE

Few = 15 to 20 on DECtape to DECtape SGEN.

Once the new system tape has been constructed, it can then be used as the installation standard tape by mounting it on the system device unit (i.e., unit 0 for DECtape), reading in the paper tape system bootstrap, etc.

APPENDIX A
DEVICE ASSIGNMENTS

Either System Generation or the ASSIGN command can be used to attach devices to the slots of the device assignment table (.DAT). The table below shows the normal setup of .DAT. Only system slots -2, -3, and -7 cannot be modified by the ASSIGN command, since these must be used by the Monitor.

System programs use the negative .DAT slots while user programs should use the positive .DAT slots. PIP-9 (Peripheral Interchange Program) is an exception to this rule in that it uses all the positive .DAT slots (1 to 10).

.DAT Slot	Device Handler		Unit	Use
	8K	16K or greater		
-15	DTA	DTA	2	Output (EDITOR, UPDATE, CONVERTER, SYSGEN)
-14	DTA	DTA	1	Input (EDITOR, UPDATE, CONVERTER, SYSGEN, DUMP)
-13	PPC	DTA	2	Output (MACRO-9, FORTRAN IV)
-12	TTA	TTA		Listing (MACRO-9, FORTRAN IV, UPDATE, DUMP, CONVERTER)
-11	DTC	DTA	1	Input (MACRO-9, FORTRAN IV)
-10	TTA	PRA		Input (DDT) Secondary Input (EDITOR UPDATE)
-7	DTC	DTC	0	System Device (System Loader)
-6	NONE	NONE		Output (DDT)
-5	NONE	NONE		External Library (Linking Loader)
-4	PRA	DTA	2	Input (Linking Loader)
-3	TTA	TTA		Teletype Output } All system programs
-2	TTA	TTA		Keyboard Input } All system programs
-1	DTC	DTA	0	System Library (Linking Loader)
1	DTA	DTA	0	User and PIP-9 .DAT slots
2	DTA	DTA	1	User and PIP-9 .DAT slots
3	DTA	DTA	2	User and PIP-9 .DAT slots
4	TTA	TTA		User and PIP-9 .DAT slots
5	PRA	PRA		User and PIP-9 .DAT slots
6	PPA	PPA		User and PIP-9 .DAT slots
7	DTA	DTA	1	User and PIP-9 .DAT slots
10	DTA	DTA	2	User and PIP-9 .DAT slots

APPENDIX B
PDP-9 ASCII CHARACTER SET

Listed below are the ASCII characters interpreted by the PDP-9 Monitor and system programs as meaningful data input or as control characters.

	00-37 ASCII CHAR.	40-77 ASCII CHAR.	100-137 ASCII CHAR.	140-177 ASCII CHAR.	
0	NUL	SP	\		0
1	SOH (A)		A		1
2		"	B		2
3	ETX (C)	#	C		3
4		$	D		4
5		%	E		5
6		&	F		6
7		'	G		7
10		(	H		10
11	HT	)	I		11
12	LF	*	J		12
13	VT	+	K		13
14	FF	,	L		14
15	CR	-	M		15
16		.	N		16
17		/	O		17
20	DLE (P)	0	P		20
21	(Q)	1	Q		21
22	DC2 (R)	2	R		22
23	DC3 (S)	3	S		23
24	DC4 (T)	4	T		24
25	NACK (U)	5	U		25
26		6	V		26
27		7	W		27
30	CNCL (X)	8	X		30
31		9	Y		31
32	SS (Z)	:	Z		32
*33	ESC	;			33
34		<			34
35		=		ESC	35
36	RS ( )	>	^ or ↑	ESC	36
37		?		delete (RO)	37

*Codes 33, 175, and 176 are interpreted as ESC (ALT Mode) and are converted on input to code 175 by IOPS handlers.

APPENDIX C
MACRO-9 ERROR DIAGNOSTICS

Flag	Meaning
A	Error in direct Symbol Table assignment, assignment ignored.
B	Memory Bank error.
D	The statement contains a reference to a multiply defined symbol. It is assembled with the first value defined.
E	Erroneous results may have been produced. Will also occur on undefined .END value.
I	Line ignored. (Redundant Pseudo-op)
L	Literal phasing error.
M	An attempt is made to define a symbol which has already been defined. The symbol retains its original value.
N	Error in number usage.
P	Phase error. PASS1 value does not equal PASS2 value of a symbol. PASS1 value will be used.
Q	Questionable line.
R	Possible relocation error.
S	Symbol error. An illegal character was encountered and ignored.
U	An undefined symbol was encountered.
W	Line overflow during macro expansion.
X	Illegal usage of macro name.

APPENDIX D
MACRO-9
PERMANENT SYMBOL TABLE

Memory Reference

CAL	000000
DAC	040000
JMS	100000
DZM	140000
LAC	200000
XOR	240000
ADD	300000
TAD	340000
XCT	400000
ISZ	440000
AND	500000
SAD	540000
JMP	600000

Operate

OPR	740000
NOP	740000
CMA	740001
CML	740002
OAS	740004
RAL	740010
RAR	740020
HLT	740040
XX	740040
SMA	740100
SZA	740200
SNL	740400
SML	740400
SKP	741000
SPA	741100
SNA	741200
SZL	741400
SPL	741400
RTL	742010
RTR	742020
CLL	744000
STL	744002
CCL	744002
RCL	744010
RCR	744020
CLA	750000
CLC	750001
LAS	750004
LAT	750004
GLK	750010
LAW	760000

EAE Type KE09A

EAE	640000
OSC	640001
OMQ	640002
CMQ	640004
DIV	640323
NORM	640444
LRS	640500
LLS	640600
ALS	640700
LACS	641001
LACQ	641002
ABS	644000
DIVS	644323
CLQ	650000
FRDIV	650323
LMQ	652000
MUL	653122
IDIV	653323
FRDIVS	654323
MULS	657122
IDIVS	657323
NORMS	660444
LRSS	660500
LLSS	660600
ALSS	660700
GSM	664000

I/O States

IOT	700000
IORS	700314

Interrupt

IOF	700002
ION	700042
CAF	703302

Automatic Priority Interrupt Type KF09A

DBK	703304
DBR	703344
SPI	705501
ISA	705504

Memory Extension Control Type KE09B

SEM	707701
EEM	707702
LEM	707704

Memory Protect Type KX09A

MPSK	701701
MPLU	701702
MPLD	701704
MPEU	701742

APPENDIX E
EXPLANATION OF IOPS ERROR CODES

ERROR CODE	ERROR	ERROR DATA
0	Illegal Function CAL	CAL address
1	CAL * illegal	CAL address
2	.DAT slot error	CAL address
3	Illegal interrupt	I/O status register
4	Device not ready (type control R when ready)	
5	Illegal .SETUP CAL	CAL address
6	Illegal handler function	
7	Illegal data mode	CAL address
10	File still active	CAL address
11	SEEK/ENTER not executed	CAL address
12	Unrecoverable DECtape error	DECtape status register B and Unit No.
13	File not found	CAL address
14	Directory full	CAL address
15	DECtape full	CAL address
16	Output buffer overflow	CAL address
17	Too many files for handler	CAL address
20	Disc failure	Disc status register
21	Illegal disc address	Illegal address
22	Two output files on one unit	CAL address
27	Illegal disc unit	CAL address
30	API software level error	API status register
31	Non-existent memory reference	Program counter
32	Memory protect violation	Program counter
33	Memory parity error	Program counter

APPENDIX F
REVISED FORTRAN IV ERROR LIST

These letter-coded error messages apply to F4 V2A and all versions of F4 thereafter and all versions of F4A.

Error Code		Cause
X	Syntax error	Statement cannot be recognized as a properly constructed FORTRAN IV statement.
V	Variable/constant mode error	Illegal mode mixing. Missing constant, variable or exponent, or illegal matching of constants or variables in a DATA statement.
N	Statement number error	Phase error, number more than 5 digits, no statement number where one is required, statement should not be labeled or doubly defined statement numbers.
S	Argument/subscript error	Missing argument or subscript, illegal use of subscripts, illegal construction of subscripted variable, more than 3 subscripts or stated number of subscripts does not agree with declared number
F	FORMAT statement error	Illegal FORMAT specification or illegal construction of FORMAT statement.
I	Character/statement/ term error	Illegal character, unrecognizable statement, illegal statement for program type, statement out of order or improper statement preceding END statement.
D	DO loop error	Illegal DO construction or illegal statement terminating DO LOOP.
T	Table overflow	Symbol/constant/arg (I)/OP(I) table limits exceeded.
L	Nesting error	Illegal nesting or DO nesting too deep.
M	Magnitude error	Program exceeds 8190 words, maximum number of dummy arguments or EQUIVALENCE classes exceeded, or constant/variable exceeds specified limits.
C	COMMON/EQUIVALENCE/ DIMENSION/DATA statement error	Illegal construction of statement, illegal EQUIVALENCE relationships, illegal COMMON declaration or noncommon storage declared in BLOCK DATA subprogram.
E	FUNCTION/SUBROUTINE/ EXTERNAL/CALL statement error	Illegal use of FUNCTION/SUBROUTINE name, out of order, or illegal variable for EXTERNAL declaration.
H	Hollerith error	Hollerith data illegal in this statement or illegal of Hollerith constant.

APPENDIX G
FORTRAN IV ERROR LIST

This list of error messages applies of all versions of F4 prior of V2A.

Error Code	Cause
001	Improper statement preceding END statement.
002	FUNCTION/SUBROUTINE name not used or used improperly in a subprogram.
003	Variable required.
004	Positive nonzero constant required.
005	Symbol-constant table limits exceeded.
006	Statement number has more than 5 digits.
007	Unsigned simple integer argument required.
008	Integer value greater than $(2^{17} - 1)$.
009	Magnitude of number (ignoring decimal point and/or exponent) greater than $(2^{35} - 1)$.
010	Array element (function reference) and next argument not separated by comma.
012	Exponent missing from numeric field.
014	Open parenthesis in subscript.
015	Binary operator used in unary sense or missing argument.
016	Additional grouping parenthesis not allowed.
017	Subscript list terminated before last argument obtained.
018	Too many right parentheses.
019	Argument follows converted argument (no operator separating them).
021	E or D in numeric field after exponent has been processed.
023	Format descriptor character used in nonformat statement.
024	Hollerith data illegal this statement.
025	Non-integer constant precedes Hollerith constant.
026	Length of Hollerith constant exceeds range $(0 < x < 6)$.
027	Hollerith constant contains statement termination character.
030	Symbolic name exceeds 6 characters.
031	Limit on number of nested functions exceeded.
032	Simple variable delineated by left parenthesis.
033	Subscripted variable used as subscript.
034	Unrecognizable logic term.
036	Logical .NOT. used as binary operator or logical constant used as a binary operator.
037	Illegal character.
040	Adjusted floating point exponent exceeds 76.

Error Code	Cause
045	Subscript expression not delimited by ", " or ")".
050	Arg (I), op (I) table overflow.
051	Unsubscripted array reference used as a variable.
052	Function name used as a variable.
053	Comma used other than as parameter separator.
054	Function or array name used in an expression representing a function parameter.
055	Expression used as an assignment variable.
056	Expression contains uneven number of parentheses.
060	Illegal mode mixing.
063	Signed assignment variable.
069	Doubly defined statement numbers or phase error.
070	Illegal statement for BLOCK DATA program.
071	Statement not a DO or DO illegal as True statement for logical IF.
072	Illegal logical IF true statement.
074	Unrecognizable statement (first 3 characters).
075	Unrecognizable statement (4-n characters).
076	Statement out of order.
077	Statement should not be labeled.
078	First character following READ or WRITE not a left parenthesis.
079	Illegal format specification in READ or WRITE.
080	Binary WRITE has no list.
081	Illegal list element separator.
082	Illegal implied DO construction.
086	DO nesting too deep.
087	Missing DO parameter.
088	Illegal DO parameter construction.
089	Illegal statement type terminated DO.
090	Improper DO nesting.
091	Illegal character following I/O unit specification.
092	Illegal character following FORMAT specification.
095	Name previously appeared in a specification statement.
096	Statement improperly delimited.
097	Illegal array declaration - DIMENSION statement.
098	Array size greater than 78192 - DATA specification statement.
100	Improper subscript list delimiter in a DIMENSION statement.
101	More than 3 subscripts (dimensions).
102	Integer argument not a constant.
103	Integer constant not less than 8192.

Error Code	Cause
104	Integer constant negative.
105	FORMAT statement has no statement number.
106	Illegal FORMAT specification.
114	COMMON block name has illegal delimiter (not /).
115	Name declared as COMMON block name previously declared otherwise.
116	COMMON block size exceeds 8190 words.
117	Declared COMMON variable is a dummy, function, or already in COMMON.
118	ASSIGN statement missing "TO."
120	Computed GOTO statement number list members not delimited.
121	Computed GOTO statement number list not delimited.
123	Assigned GOTO statement number list not preceded by left parenthesis.
124	Assigned GOTO statement number list not delimited.
126	DATA/EQUIVALENCE variable is a dummy variable.
127	DATA/EQUIVALENCE variable is a function name.
128	DATA/EQUIVALENCE simple variable subscripted by more than one number.
129	DATA/EQUIVALENCE array element greater than 8192.
130	Missing right or redundant left parenthesis in FORMAT statement.
131	Stated number of subscripts does not agree with declared number.
132	EQUIVALENCE class not started with left parenthesis.
136	EQUIVALENCE class improperly delimited (no right parenthesis).
137	Maximum number of EQUIVALENCE classes exceeded.
138	Illegal EQUIVALENCE relationships.
139	Illegal extension of common block caused by EQUIVALENCE relationship.
140	Illegal variable for EXTERNAL declaration.
142	Program size exceeds 8190 words.
143	Non-common storage declared in BLOCK DATA subroutine.
144	IF expression not terminated by closing parenthesis.
145	No comma separating statement numbers - IF statement.
147	No statement number where one is required.
148	Referenced statement number does not appear as a statement label.
149	Illegal statement function name (used twice or is external).
150	Delimiter following statement function dummy argument list not "=".
151	Function statement out of order (or more than one).
152	FUNCTION name not followed by argument list.
153	Dummy variable previously declared as something else.
154	Dummy argument list not enclosed in parenthesis.
155	Maximum number of dummy arguments exceeded.
156	STOP/PAUSE statement constant contains a non-octal digit.

Error Code	Cause
157	STOP/PAUSE statement constant contains too many digits.
158	DATA statement variable is "common" but data not "block data."
159	DATA statement is "block data" but variable is not "common."
160	DATA statement variables not delimited by a slash.
161	DATA statement argument not a constant.
162	Mode of variable and corresponding constant disagree in a DATA statement.
163	DATA statement constants not separated by a comma (may indicate more variables than constants).
164	Two successive asterisks used as operators in a DATA statement.
165	Constant preceding asterisk is not integer in a DATA statement.
167	More constants than variables in a DATA statement.
168	RETURN statement in main-body program unit (not subroutine).
169	CALL statement - name is not a function name.

APPENDIX H
FORTRAN IV OTS ERRORS

Error Number	Error Description	Library Routines* That May Cause Error
00-04	Not used	
05	Negative REAL Square Root Argument	SQRT
06	Negative DOUBLE PRECISION Square Root Argument	DSQRT
07	Illegal Index in Computed GO TO	.GO
10	Illegal I/O Device Number	.FR,.FW,.FS,.FX, .FR,.FA,.FE,.FF,.FS,
11	Bad input data - IOPS Mode Incorrect	.FR,.FA,.FE,.FF,.FS,
12	Bad FORMAT	.FA,.FE,.FF
13	Negative or Zero REAL Logarithmic Argument	.BC,.BE,ALOG
14	Negative or Zero DOUBLE PRECISION Logarithmic Argument	.BD,.BF,.BG,.BH, DLOG,DLOG10

* Only those routines whose calls are generated by the compiler are listed.

APPENDIX I
KEYBOARD MONITOR ERRORS

Errors	Explanation
WHAT?	Unrecognizable command
BAD DEV - IGNORED FROM ERR	Illegal device reference, for example: A PRA 5,6/PPW7/DTA-5 where the command is processed and effective up to the PPW and the remainder of the command is ignored.
BAD .DAT SLOT - IGNORED FROM ERR	Illegal .DAT slot reference, for example: A PRA 5,6/PPA G where the command is processed and effective through A PRA 5,6 but ignored from there on.
BAD PRGNAM	Non-existent program name. Command ignored.
PERMANENT .DAT SLOT - IGNORED FROM ERR	Command attempted to assign a device handler to one of the permanent .DAT slots (-2, -3, or -7).
BAD UNIT - IGNORED FROM ERR	Illegal unit reference (e.g., DTAX)
BAD START LOC	Illegal address given in "GET n address" command.
SYS DEV ERR - CHECK UNIT & TRY AGAIN	Last command typed caused error condition on system device control.
BAD COMMAND IN BATCH MODE	Illegal Batch Processor command: QDUMP, HALT, GET (all forms), BATCH, LOAD, DDT, or DDTNS.
BAD BATCH DEV	Batch device was not designated properly. Should be: CD - for card reader PR - for paper tape reader
BAD $JOB COMMAND	$JOB command not terminated by space, carriage return, or ALT MODE.

APPENDIX J
SAMPLE PROGRAM USING SYSTEM SOFTWARE

PDP-9 ADVANCED Software System can best be illustrated by solving an actual problem. A FORTRAN program to compute solutions of quadrative equations is used as an example on the following pages. Vertical spacing has been added at some points to facilitate annotation. Using the Editor to type in the program and FORTRAN IV to compile, the program can be typed and compiled very quickly. Using DDT to load and execute the program allows for dynamic debugging if required. The system tape (8K system) was loaded on DECtape unit 0 (8) and the system bootstrap was loaded using the hardware read-in mode prior to the beginning of the example. The example begins with the Monitor typing

```
MONITOR
$
```

on the teleprinter to indicate that it has been loaded and is ready to receive user command.

```
MONITOR

$R EDIT

 .DAT      DEVICE     USE

  -15       DTA2      OUTPUT/SCRATCH
  -14       DTA1      INPUT/OUTPUT
  -10       TTA0      SECONDARY INPUT
  -3        TTA0      TELEPRINTER OUTPUT AND ERRORS
  -2        TTA0      COMMAND STRING
```

REQUEST (R) allows user to check Editor .DAT slot assignments for possible modification.

```
$A DTA0 -15,-14
```

ASSIGN (A) allows user to reassign preset .DAT slots. Example shows .DAT slots -14 and -15 assigned DECtape unit 0 (8).

```
$EDIT
```

EDIT calls Editor into core.

```
 EDITOR
>OPEN QUAD
FILE QUAD SRC NOT FOUND.
INPUT
C
C          QUADRATIC EQUATION SOLUTION
C
C          READ INPUTS: A,B,C AND PRINT
C
50         READ(5,100)A,B,C
100        FORMAT(3F10.3)
           WRITE(6,101)
101        FORMAT(21H THE INPUTS A,B,C ARE)
           WRITE(6,100)A,B,C
```

EDITOR is typed to indicate that the Editor has been loaded and is ready to receive a command. User types OPEN QUAD. The Editor searches for a file named QUAD, and when it is not found, it creates a new file, names it QUAD, and changes to INPUT mode. The user types his FORTRAN program.

```
C
C           SOLVE QUADRATIC
C
            D=B**2-4.Ø*A*C
            RD=SQRT(D)
            X1=(-B+RD)/(2.Ø*A)
            X2=(-B-RD)/(2.Ø*A)
C
C           PRINT SOLUTION
C
            WRITE (6,2ØØ)
2ØØ         FORMAT(18H THE SOLUTIONS ARE)
            WRITE(6,2Ø1)X1,X2
2Ø1         FORMAT(2F1Ø.3)
            GO TO 5Ø
            END
```

```
EDIT
>CLOSE
```

User types a carriage return as the first character of the line. The Editor returns to EDIT mode, and the user types CLOSE to close his file.

```
 EDITOR
>↑C
```

EDITOR is typed to indicate that the Editor is ready for another command. The user types ↑C to return to the Monitor.

```
MONITOR

$R F4

 .DAT     DEVICE    USE

  -13      PPCØ     OUTPUT
  -12      TTAØ     LISTING
  -11      DTC1     INPUT
  -3       TTAØ     CONTROL AND ERROR MESSAGES
  -2       TTAØ     COMMAND STRING
```

MONITOR is typed to indicate that control has returned to the Monitor. The user types R F4 to check .DAT slot assignments for the FORTRAN compiler.

```
$A DTCØ -11

$F4
```

The user assigns DTC0 to .DAT slot -11 for input to the compiler, and then calls the compiler by typing F4.

```
FORTRAN 4
>B,S,L←QUAD,
```

The compiler types FORTRAN 4 when it has been loaded and is ready to receive a command. The user types B,S,L←QUAD,

to initiate compilation with option as follows.

B = Binary output
S = Symbol table listing
L = Source listing

```
END PASS1
```

END PASS1 indicates the compiler has completed the first pass. Any compiler errors would have been typed out before this message.

Source listing of user program.

```
C
C           QUADRATIC EQUATION SOLUTION
C
C           READ INPUTS: A,B,C AND PRINT
C
50          READ(5,100)A,B,C
100         FORMAT(3F10.3)
            WRITE(6,101)
101         FORMAT(21H THE INPUTS A,B,C ARE)
            WRITE(6,100)A,B,C
C
C           SOLVE QUADRATIC
C
            D=B**2-4.0*A*C
            RD=SQRT(D)
            X1=(-B+RD)/(2.0*A)
            X2=(-B-RD)/(2.0*A)
C
C           PRINT SOLUTION
C
            WRITE (6,200)
200         FORMAT(18H THE SOLUTIONS ARE)
            WRITE(6,201)X1,X2
201         FORMAT(2F10.3)
            GO TO 50
            END
```

Symbol table listing.

```
  QUAD    17777
  .50     00000
* .FR     00202
  .100    00012
  A       00221
* .FE     00203
  B       00223
  C       00225
* .FF     00204
* .FW     00205
  .101    00023
  D       00227
* .AG     00206
```

```
* .BC     00207
  .B1     00231
* .AH     00210
* .AK     00211
* .AM     00212
  RD      00233
* SQRT    00213
  X1      00235
* .AJ     00214
* .AN     00215
  X2      00237
* .AI     00216
* .BA     00217
  .200    00140
  .201    00163
* .FP     00220
```

```
FORTRAN 4
>↑C
```

Compiler types FORTRAN 4 to indicate that it is ready for more input. User types ↑C to return to Monitor.

```
MONITOR

$R DDT

 .DAT      DEVICE     USE

  -10       TTA0      PATCH INPUT
  -6        NONE      PATCH OUTPUT
  -5        NONE      USER LIBRARY
  -4        PRA0      USER PROGRAM(S)
  -3        TTA0      TELEPRINTER OUTPUT AND ERRORS
  -2        TTA0      COMMAND STRING
  -1        DTC0      SYSTEM LIBRARY & DDT
```

MONITOR is typed to indicate that control has returned to the Monitor. The user types R DDT to check .DAT slot assignments for DDT.

```
$A TTA0 5,6
```

User assigns Teletype for input and output as indicated in his FORTRAN source program.

```
$DDT
```

User places his binary object tape in the paper tape reader and types DDT to load and execute his program.

```
LOADER

>QUAD
 QUAD    14337
 .BC     14273
 SQRT    14205
 .EE     14114
 .EF     13776
 .EC     13732
 .DA     13663
 BCDIO   10670
 FIOPS   10134
 OTSER   10026
 REAL    07073
```

LOADER is typed to indicate that DDT has been loaded. The user types QUAD (terminated by ALT MODE) to initiate loading. The Linking Loader (which is part of DDT) types a memory map for the object program and all FORTRAN library subroutines used.

```
DDT
>'
```

DDT is typed to indicate that it is now in control and ready to receive a command. The user types an apostrophe (') and execution of his program begins.

```
3.000          6.000          3.000
THE INPUTS A,B,C ARE
    3.000          6.000          3.000
THE SOLUTIONS ARE
   -1.000         -1.000
6.000          24.000         6.000
THE INPUTS A,B,C ARE
    6.000         24.000         6.000
THE SOLUTIONS ARE
   -0.268         -3.732
↑T
```

The user types three variables (A, B, and C as required by his program). The program repeats the inputs along with the solutions.

```
DDT
```

User types ↑T to return control to DDT.

```
>A/    CAL+3     :    000003

>B-1/    ADD     :    300000
 B/    CAL+5     :    000005
 C-1/    ADD     :    300000
 C/    CAL+3     :    000003
 D-1/    ADD
 D/    CAL+12011     :    012011

>↑C
MONITOR

$
```

User examines contents of locations containing variables and finally types ↑C to return control to the Monitor.

APPENDIX K
EXAMPLE OF BATCH PROCESSING

The following example was produced under control of the Batch Processor. Underlined commands are on paper tape. ALT MODE termination is indicated with a .

```
MONITOR V3A
$BATCH PR                                   This command causes all subsequent commands to
MONITOR V3A                                 come from the paper tape reader.
$$JOB TEST BATCH
$PIP
PIP V4A
>N DT1
>T DT1 TEST SRC (A) ← PR⊗
$DATA
$END                                        The entire program to be compiled below appears on
MONITOR V3A                                 the paper tape between $DATA and $END.
$$JOB
$R F4
 .DAT    DEVICE    USE
 -13      DTA2     OUTPUT
 -12      TTA0     LISTING
 -11      DTA1     INPUT
 -3       TTA0     CONTROL AND ERROR MESSAGES
 -2       PR*0     COMMAND STRING
$F4
FORTRAN 4 V2A
>S,L,B←TEST ⊗
END PASS1
C
C BATCH PROCESSOR TEST                      FORTRAN prgram to list numbers from 1 to 10
C
              DO 1 I=1,10
1             WRITE (4,100) I
100           FORMAT (6X,I3)
              STOP 12345
              END
```

```
   TEST   17777
   .1     00012
   I      00043
 * .FW    00036
   .100   00021
 * .FE    00037
 * .FF    00040
 * .ST    00041
 * .FP    00042

MONITOR V3A
$$JOB
$GLOAD

LOADER V2A
>TEST ⊗
 TEST    37734
 BCDIO   34741
 STOP    34726
 SPMSG   34632
 FIOPS   34076
 OTSER   33770
 REAL    33035
```

Program execution begins here.

```
          1
          2
          3
          4
          5
          6
          7
          8
          9
         10
STOP 012345
MONITOR V3A
$$JOB
$$EXIT

MONITOR V3A
$
```

Control is returned to Teletype at this point.

digital

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

Printed in U.S.A.