CONTENTS

FORTRAN-4

MACRO-9

PIP-9

EDITOR

INTRODUCTION

This guide for operating the paper tape version of the PDP-9 ADVANCED Software System is planned for convenient use at the computer. It contains general operating instructions, as well as concise summaries of operating procedures for individual system programs. For more detailed descriptions of these programs, the reader is referred to the PDP-9 Monitor Programmer's Reference Manual (DEC-9A-MABO-D) and the Utility Programs Manual (DEC-9A-GUAB-D).

1. LOADING PROGRAMS

In the paper tape system, each system program, accompanied by the necessary I/O device handlers and an appropriate version of the I/O Monitor, resides on a separate paper tape in absolute format. The eight system tapes supplied are:

FORTRAN IV
MACRO-9
PIP-9
Editor (EDIT-9)
Linking Loader (LINK-9)
DDT-9 (without patch file capabilities)
DDT-9 (with patch file capabilities)
7-TO-9 Converter (CONV-9)

To load these programs, place the tape in the reader, set the loading address in the console address switches, press the tape feed button, depress I/O RESET, and then depress the READIN switch.

The loading addresses are:

17720 for 8K systems
37720 for 16K systems
57720 for 24K systems
77720 for 32K systems

Either the Linking Loader or DDT-9 may be used to load user programs.

2. SYMBOLS (Used In This Manual)

↵	Carriage return
␣	Space
↑	Represents the CTRL key on the Teletype
(CTRL)	Represents the CTRL key on the Teletype
(ALT MODE)	Represents the ALTMODE key on the Teletype

1. LOADING INSTRUCTIONS

Put the paper tape FORTRAN IV Compiler in the reader, depress the tape-feed control to clear the end-of-tape flag, set the address switches to 17720 (8K), depress I/O RESET and then READ IN.

When FORTRAN IV is ready to accept a command string (on the same line as the >) it prints on the teleprinter:

```
FORTRAN 4
>
```

2. GENERAL COMMAND CHARACTERS

RUBOUT (echos \ )	delete single character
CTRL U (↑U) (echos @)	delete entire line
CTRL P (echos ↑P)	a) at end of pass 1, begin pass 2 b) while compiler is running restart at beginning of pass 1

3. COMMAND STRING

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

Options		File Name	Terminator
O, S, L, B	←	FILEX,	↲ or ALT·MODE

where

FILEX — must be a legal FORTRAN name

, — file name terminator (required)

O = object listing

S = symbol map

L = source listing

B = binary

ALT·MODE — halts after compiling current program

↲ — indicates batch compilation; after compiling current program, types

```
FORTRAN 4
>
```

and waits for next command string.

The 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 none of the options are desired, ← is sufficient, with the sole output being compiler diagnostics on the teleprinter. The file name must be terminated by a comma. Rubouts may be used to delete unwanted characters, and control U (↑U) to delete entire lines, prior to typing the command string terminator.

4. RUNNING INSTRUCTIONS

After the compiler is loaded into core,

a. Put the source program tape in the paper tape reader, momentarily depressing the tape-feed control.

b. Type the command string.

c. At the end of Pass 1 (when the END statement is encountered for the first time), FORTRAN IV indicates:

```
END PASS 1
↑ P
```

d. Reload the source tape for Pass 2, momentarily depressing the tape-feed control.

e. Initiate Pass 2 by typing control P (↑P).

5. ERROR CONDITIONS AND RECOVERY

IOPS 4 Device not ready. Check devices, correct condition, and type Control R (R).

IOPS 00-30 See SYSTEM RESTART procedures.

See Appendix 6 for a list of compiler errors.

6. RESTART PROCEDURES

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

See Appendix 5 for SYSTEM RESTART operation if FORTRAN IV has halted.

7. 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.

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,↲
```

1. LOADING INSTRUCTIONS

Put the paper tape MACRO-9 assembler in the reader, depress the tape-feed control to clear the end-of-tape flag, set address switches to 17720(8K), depress I/O RESET and then READ IN. When MACRO-9 is ready to accept a command string (on the same line as the >), it prints on the teleprinter:

MACRO
>

2. GENERAL COMMAND CHARACTERS

RUBOUT (echos \\)	delete single character
CTRL U (↑) (echos@)	delete complete line
CTRL P (echos↑ P)	a) at end of pass 1, begin pass 2
	b) while assembler is running, restart at beginning of pass 1

3. COMMAND STRING

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

Options	File Name	Terminator
P, S, L, B	←FILEX	↵ or ALT MODE

halts after assembling current program

where

B = binary

L = listing

S = symbol table (on listing device)

P = parameters to be entered on teletype

return to MACRO-9 after assembling current program, types

MACRO
>

and waits for next assembly command string.

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 desired, ← is sufficient and the sole output will be assembly error messages on the teleprinter. Rubouts may be used to delete unwanted characters, and control U (↑U) to delete entire lines, prior to typing the command string terminator.

4. RUNNING INSTRUCTIONS

After the assembler is in core,

a. Load the user program source tape in the paper tape reader, momentarily depressing the tape-feed control.

b. Type the command string.

c. At the end of Pass 1 (when the .END statement is encountered for the first time) MACRO indicates:

```
END OF PASS 1
↑P
```

d. Reload the source tape for Pass 2, momentarily depressing the tape-feed control.

e. Initiate Pass 2 by typing control P (↑P).

If this is a multi-tape assembly (where the first n source tapes are terminated with .EOT and the last is terminated with .END), MACRO-9 indicates the end of each tape by typing .EOT on the teleprinter. This allows the user to load the next source tape (depress the tape-feed control) and then type control P (↑P).

If the P option was used, the parameters are entered only at the beginning of Pass 1 and not again for Pass 2.

5. ERROR CONDITIONS AND RECOVERY PROCEDURES

IOPS 4 Device not ready. Check devices, correct condition and type Control R (↑R).

IOPS 00-30 See SYSTEM RESTART procedures.

6. RESTART PROCEDURES

CTRL P (↑P) Restart MACRO-9, if running.

See Appendix 5 for operation of SYSTEM RESTART if MACRO-9 has halted.

7. 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 Teletype, type the command string:

```
P, B←FILEX↵
```

The parameters should be entered during the first pass only. Parameters are typed following the command string, in the form of MACRO-9 direct assignment statements. After typing in parameters, the user types ↑D, as shown in the example below.

BANK = 0

CTRL D (EOT)

MACRO the outputs

EOT

↑P

type control P (↑P) when ready to proceed.

3. If the output desired is a complete listing but no binary tape, type the command string:

S, L←FILEX ↵

1. LOADING INSTRUCTIONS

Put the paper tape PIP program in the reader, depress the tape-feed control to clear the end-of-tape flag, set the address switches to 17720 (8K), depress I/O RESET and then READ IN.

When PIP-9 is ready to accept a command string (on the same line as the >), it prints on the teleprinter:

```
PIP
>
```

2. GENERAL COMMAND CHARACTERS

RUBOUT (echos \\)	delete single character
CTRL U (↑) (echos@)	delete entire line
CTRL P (echos↑P)	restart PIP

3. COMMAND STRING

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

F DD (S) ← SD

terminated by a carriage return or ALT MODE.

F is a function character, which may be:

T = transfer file

V = verify file

S = segment file

DD is the destination device.

PP = paper tape punch

TT = teletype

LP = line printer

(S) indicates the switch options.

Data Mode Switches:

A = IOPS ASCII

B = IOPS binary

I = Image Alphanumeric

Function switches:

G = correct bad parity lines

E = convert tabs to spaces

C = convert multiple spaces to tabs

Y = segment files (with n output tapes, use n-1 commas after PP)

W = combine files (with n input tapes, use n-1 commas after PR)

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

SD is the source device.

PR = paper tape reader

TT = teletype

CD = card reader

Carriage Return or ALT MODE is the command string terminator:

Carriage Return - return to PIP after completion of the current function.

ALT MODE - halt after completion of the current function.

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

4. OPERATING INSTRUCTIONS

Legal function/switch combinations

Transfer (T):	all switches legal
ASCII mode (A):	all function switches legal E and C are contradictory Y and W are contradictory
Binary mode (B):	function switch W only
Image mode (I):	no function switches legal
Verify (V):	switches A and B only
Segment (S):	no switches legal

G Switch Use

When bad parity is encountered, the INPUT PARITY ERROR message is output on the Teletype, followed by the line in error. The user may:

a. Accept the line by typing a carriage return.

b. Delete the line by typing

D↲

c. Retype the line, terminating with a carriage return.

d. Abort the operation by typing Control P (↑P) to restart PIP.

5. ERROR CONDITIONS AND RECOVERY PROCEDURES

IOPS 4 device not ready	Ready device and type control R (↑R)
IOPS 00-30	See SYSTEM RESTART Procedures

6. RESTART PROCEDURES

CTRL P (↑P) restart PIP, if running

See Appendix 5 for operation of SYSTEM RESTART if PIP has halted.

7. EXAMPLES

a. To reproduce an ASCII tape:

T␣PP␣(A)␣←␣PR↵

Transfer to the paper tape punch from the paper tape reader in IOPS ASC11 mode.

b. To list an ASCII tape:

T␣TT␣(A)␣←␣PR↵

c. To combine 3 binary subprogram tapes into one tape:

T␣PP␣(BW)␣←␣PR,,↵

Since the W switch is on, the 3 binary tapes will be combined into one file, with the intermediate EOF's deleted. The final EOF is retained. This provides a very convenient method of creating a Library file.

d. To verify a binary tape:

V␣PR␣(B)↵

Checksum and Parity verification will be performed on the input binary tape. There will be no output. If a parity error occurs, the following message is typed:

INPUT PARITY ERROR

If a checksum error:

INPUT CHECKSUM FAILURE

e. To check parity:

T␣PP␣(AG)␣←␣PR↵

Transfer files from paper tape to paper punch in ASCII mode with G switch to check the bad parity.
For actions to be taken if any bad parity is encountered, refer to the explanation for G switch.

f. To reproduce a binary tape:

T␣P␣(B)␣←␣PR↵

Transfer files from paper tape reader to paper tape punch in Binary mode.

g. To reproduce in Image mode:

```
T␣PP␣(I)␣←␣ PR↵
```

Transfer files from paper tape reader to paper tape punch in image ASCII mode. This is the only way to reproduce a tape with channel 7 punches.

h. To segment a tape:

```
S␣TAGA, TAGB, TAGC↵
```

Sets up the segmentation points.

```
T␣PP,,,(AY)←PR↵
```

Transfers from paper tape reader to paper tape punch, providing EOT and blank tape just before each indicated tag.

P is output by PIP at the end of each segment.

When ready to continue, type control P (↑P).

1. LOADING INSTRUCTIONS

Put the Editor Tape in the reader, depress the tape-feed control to clear the end-of-tape flag, set the address switches to 17720 (8K), depress I/O RESET and then READ IN. When the Editor is in core, it indicates its readiness to accept a command (on the same line as the >) by outputting on the teleprinter:

```
EDITOR
>
```

It is initially in Edit Mode.

One may either create a file or edit an existing file.

2. GENERAL COMMAND CHARACTERS

RUBOUT (echos \)	delete single character
CTRL U (↑) (echos@)	delete entire line
CTRL P (echos↑P)	restart the editor

3. COMMAND STRING

Not applicable

4. OPERATING PROCEDURES

Editing Operation 1: Creating a file

(When Editor is brought in core, it is in Edit Mode)

User types in	ACTION	EFFECT
1. ↲	INPUT	Mode is changed from Edit to Input
2. Content of the program (each line is terminated by ↲)	Punches out previous line typed.	Line typed in is processed.
3. ↲ (necessary before close)	EDIT >	Change from Input to Edit Mode.
4. CLOSE	punches blank tape EDITOR >	Finishes the current file.

Editing Operation 2: Modifying an existing file

a. Put the source tape in the reader

b. Depress tape-feed control

c. Type in any command desired. See summary of Edit commands listed below.

SUMMARY OF EDITING COMMANDS

COMMAND	ABBREVIATION	ACTIVITY
File Housekeeping Requests		
CLOSE	n/a	Terminate editing on input files.
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 to work area.
TOP*	T	Reset pointer to beginning of block.
PRINT	P	Print the current line on the Teletype.
Manipulative Requests		
MOVE_TAG1_TAG2_TAG3*_	M	Perform a block transfer of several lines in the buffer.
DELETE	D	Discard the current line.
RETYPE string	R	Replace current line with "string".
INSERT string	I	Add "string" as a complete line, after (below) the current line.
CHANGE$_u$/string1/string2/	C	Replace, in the current line, the first occurrence of "string1" with "string2"
OVERLAY	O	Replace multiple lines.
APPEND string	A	Add "string" of the rightmost end of the current line.

*May be used only with BLOCK mode ON.

COMMAND		ABBREVIATION	ACTIVITY
Mode Control			
VERIFY	{ ON OFF	V	Set verify mode to print (ON) or ignore printing (OFF) lines after processing CHANGE, LOCATE, FIND and BOTTOM requests.
BLOCK	{ ON OFF	n/a	Set program to operate in block mode (ON) or in line-by-line mode (OFF).
BRIEF	{ ON OFF	n/a	Set brief mode to print truncated (ON) or full (OFF) lines.
Input/Output Requests			
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.

5. ERROR CONDITIONS AND RECOVERY PROCEDURES

a. END OF FILE REACHED BY:

NEXT n

Results if the command results in the pointer moving past the last line of the file or buffer.

a) If editing in line by line mode, use command CLOSE and reload the input tape.
b) If in block mode, move the pointer to the top of the buffer (T↵).

b. END OF FILE REACHED BY:

PRINT n

For recovery, do exactly as above.

c. END OF MEDIUM REACHED BY:

GET n

If the end-of-medium condition is encountered on the subsidiary input device before n lines are read. The pointer remains at the last line read.

Put the original paper tape back in the reader where it left off and continue editing.

*May be used only with BLOCK mode ON.

d. NOT A REQUEST

P3↵ P␣3↵

Blank required between command and agreement.

e. NOT A REQUEST

D␣ . ↵ D↵

␣ . is not recognized by Editor as a symbol with correct counter value.

f. IOPS 4

Device not ready: ready device and type control R (↑R)

g. IOPS 00-30

See Restart Procedures.

6. RESTART PROCEDURES

CTRL (↑P) restart editor if running.

See Appendix 5 for operation of SYSTEM RESTART if EDITOR has halted

7. Examples:

Purpose	Original	Desired Change	Correct Format Command (user types in)
To change 1 character in a word	JMP TAG1	JMS TAG 1	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↵	Set up Mode
Size␣N↵	N = number of lines in block
READ↵	N lines are brought in core

WRITE ↵	output all lines on to paper tape punch
BLOCK OFF↵	back to line by line editing

1 LOADING INSTRUCTIONS

Put the paper tape Converter in the reader, push the tape-feed control to clear the end-of-tape flag, set the address switches to 17720 (8K), depress I/O RESET and then READ IN.

When the Converter is ready to accept a command string, it types:

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

2. GENERAL COMMAND CHARACTERS

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

3. COMMAND STRING

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

Options	File Name	Terminator
L,A, R, E, Tn	← FILE ↳ output program name	↵ or ALT mode ↳ Halt after converting.

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

↵ ↓ return to the converter to perform more conversions, types
7-TO-9 CONVERTER
>
and waits for next string

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 to delete entire lines, control U (↑U) prior to typing the command string terminator. If an error in the command string is detected, the converter types:

```
COMMAND STRING ERROR
>
```

and waits for a new command string.

4. OPERATING INSTRUCTIONS

The input tape to be converted must be ready in the reader (push the tape-feed control to clear the end-of-tape flag) before the command string terminator is typed. When the end of the input tape is reached, the converter will punch a few inches of blank tape, then do one of the following:

a. If a carriage return was used in the command string, the converter re-initializes, types out

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

and waits for another command string.

b. If an ALT MODE was used in the command string, the computer halts. If the Tn option has been used to combine a number of tapes into one tape, the converter will type↑P at the end of each tape except the last. Place the next tape in the reader, push the tape feed control to clear the end-of-flag and type control P. Note that the Converter does not punch any blank tape at this time.

5. ERROR CONDITIONS

COMMAND STRING ERROR	Retype command string.
IOPS 4	Device not ready, possibly punch out of tape, make device ready and type control R (↑R)
IOPS 00 - 30	See SYSTEM RESTART procedure

6. RESTART PROCEDURE

CTRL P (↑P) reinitialize

See Appendix 5 for operation of SYSTEM RESTART if the converter has halted.

7. EXAMPLES

To convert a single paper tape to be assembled in the absolute mode with no listing, the command string would be

```
A←NAME↵
```

To combine three tapes into one ending with .EOT, to be assembled relocatably (but with locations settings) and to obtain a listing; the command string would be:

```
L, E, T3←NAME↵
```

1. LOADING PROCEDURE

Place the paper tape Linking Loader in the reader, push the tape feed control to clear the end-of-tape flag, set the address switches to 17720 (8K), depress I/O RESET, then READ IN. When the loader is ready, it types:

```
LOADER
>
```

2. GENERAL COMMAND CHARACTERS

RUBOUT	Delete last character typed. n rubouts may be used to delete n characters within a program name.
CTRL P	continue loading
CTRL S	start user's program

3. COMMAND STRING

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

```
>NAME1, NAME2, NAME3 Alt Mode

>NAME1 ↵
>NAME2 ↵
>NAME3 Alt Mode

>,, Alt Mode
```

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.

4. OPERATING PROCEDURES

Place the main program in the reader, push the tape feed button to clear the end-of-tape flag, then enter the command string.

When the main program has been loaded, the loader will type↑P. Place any subroutines to be loaded in the reader, push the tape feed button, and type control P.

When all subroutines have been loaded, place the I/O library (tape 1 of 3) in the reader, push the tape feed button, and type control P.

MACRO programs (that do not require programs from the FORTRAN library)

If the loader has not been satisfied at the end of the I/O library, place the short EOF tape (included with library tape) in the reader, push the feed button and type control P.

FORTRAN programs (and MACRO programs that require programs from the FORTRAN library)

After the I/O library has been read, load the FORTRAN library (tapes 2 & 3) in the same manner. (Tape 3 includes EOF)

If the loader is not satisfied by the library tapes, a subroutine has been omitted. The loader types out the names and addresses of all programs and library subroutines loaded. A .LOAD 3 error message and a zero address indicates a missing subroutine. (If this happens, it is necessary to reload the LOADER.) When loading has been successfully completed, the loader types ↑S. Ready all I/O devices needed by the user program and type control S to start execution.

5. ERROR 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
.IOPS 4	Device not ready. Ready device and type ↑R (control R)
.IOPS 00 - 30	Unrecoverable I/O error

6. RESTART PROCEDURE

None available, LOADER must be reloaded.

7. EXAMPLES

```
LOADER
>EX1 (alt mode)
 EX1    17365
↑S                      Type control S to start program.
```

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

```
 LOADER
 >EX2,SUB (alt mode)
  EX2     17656
↑P↑P                      Place subroutine in reader and type control P.
  SUB     17613
↑P↑P                      Place I/O library in reader and type control P.
↑P↑P                      Place FORTRAN library tape 2 in reader and type
  .DA     17544           control P
↑P↑P                      Place FORTRAN library tape 3 in reader and type
  BCDIO   14551           control P
  STOP    14536
  SPMSG   14442
  FIOPS   13712
  OTSER   13604
  REAL    12651
↑S
```

NOTE

After placing tape in reader, push the tape feed button to clear the end-of-tape flag. First↑P is signal to load next tape. Second↑P is acknowledgement of user typing control P.

```
 LOADER
 >, (alt mode)            Program names not needed with paper tape input to loader.
  EX2↵    17656
↑P↑P
  SUB     17613
↑P↑P
↑P↑P
  .DA     17544
↑P↑P
  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     17656
  EX2
  P P
  SUB     17613
  P P
  P P
  .DA     17544
  P P
  BCDIO   14551
  STOP    14536
  SPMSG   14442
 FIOPS   13712
 OTSER   13604
 REAL    12651
↑S


LOADER

>
 EX2     17656
↑P↑P
↑P↑P
↑P↑P
 BCDIO   14663
 STOP    14650
 SPMSG   14554
 FIOPS   14024
 OTSER   13716
 REAL    12763
 SUBROT  00000                The subroutine was omitted.
 .LOAD 3                      Unsatisfied global symbol.
```

1. LOADING PROCEDURES

Place paper tape DDT (which includes the Linking Loader) in reader, push tape feed button to clear end-of-tape flag, set address switches to 17720 (8K), push I/O RESET, then READ IN. When the tape has been loaded, DDT types:

```
LOADER
>
```

DDT then waits for the user's command string to load his program to be debugged.

2. GENERAL COMMAND CHARACTERS

RUBOUT	delete last character typed (during load phase only)
CTRL P (echos ↑P)	during load phase, continues loading with new tape
CTRL T (echos ↑T)	restart DDT or bypass loading

3. 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
>NAME1↲                           , , Alt Mode
>NAME2↲      Alt Mode
>NAME3
```

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.

4. OPERATING PROCEDURES

Place the main program in the reader, push the tape feed button to clear the end-of-tape flag, then enter the command string.

When the main program has been loaded, the loader will type↑P.

Place any subroutines to be loaded in the reader, push the tape feed button and type control P (↑P).

When all subroutines have been loaded, place the I/O library (tape 1 of 3) in the reader, push the tape feed button, and type control P (↑P).

MACRO programs (that do not require programs from the FORTRAN library)

If the loader has not been satisfied at the end of the I/O library, place the short EOF tape (included with library tape) in the reader, push the feed button, and type control P.

FORTRAN programs (and MACRO programs that require programs from the FORTRAN library)

After the I/O library has been read, load the FORTRAN library (tapes 2 & 3) in the same manner (tape 3 includes EOF).

If the loader is not satisfied by the library tapes, a subroutine has been omitted. The loader types out the names and addresses of all programs and library subroutines loaded. A .LOAD 3 error message and a zero address indicates a missing subroutine. (If this happens, it is necessary to reload DDT.)

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, see the DDT manual.

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
"	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	(control Z) Close the location, open addressed location, continue original sequence

↑A	(control A) Close the location, open addressed location, start new sequence.
↑X	(control 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 (15-bit addresses)
SYM$	Type contents as symbolic instructions (assumed if unspecified)
:	Retype in alternate mode (NUM$, SYM$)
=	Retype as transfer vector
REL$	Type addresses as relative to defined symbols (assumed if unspecified)
RLC$	Type addresses as relocatable numbers
ABS$	Type addresses as absolute numbers

Starts and Restarts

'	Starts user's program at normal starting point
k'	Starts user's program at location k
	Restarts user's program from breakpoint
n'	Restarts 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

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
k␣PFO$	Single location k patch file output
SN$$	Save new symbols
PFE$	Close patch file output

Patch File Input

PFI$	Read patch file

Coresident Subroutines

kHDR$	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

ERROR CONDITIONS

a. loader errors

.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

b. 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 indicator - current entry ignored undefined symbol address above core incorrect command illegal character

c. I/O errors

.IOPS 4 device not ready - ready device and type control R (↑R)

.IOPS 00 - 30 unrecoverable during loading phase, returns to DDT during debug phase,

6. RESTART PROCEDURE

CTRL T (↑T) Restarts DDT

If halted, DDT must be reloaded.

7. EXAMPLES

```
 LOADER

 >EX1 (alt mode)
  EX1     14455


 DDT

 >


 LOADER

 >EX2, SUB (alt mode)
  EX2     14746
↑P↑P                    Place subroutine in reader and type control P.
  SUB     14703
↑P↑P                    Place I/O library in reader and type control P.
↑P↑P                    Place FORTRAN library tape 2 in reader and type control P.
  .DA     14634
↑P↑P                    Place FORTRAN library tape 3 in reader and type control P.
  BCDIO   11641
  STOP    11626
  SPMSG   11532
  FIOPS   11002
  OTSER   10674
  REAL    07741
```

NOTE

After placing tape in reader, push the tape feed button to clear the end-of-tape flag. First ↑P is signal to load next tape. Second ↑P is acknowledgment of user typing control P.

```
 DDT

 >


 LOADER

 > (alt mode)                   Program name not needed with paper tape input to loader.
  EX1          14455
```

```
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

>SYM$
>./   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

>
```

APPENDIX 1

DEVICE ASSIGNMENTS

Device Assignment Tables (.DAT)

In the I/O Monitor version, the .DAT slot assignments are permanent and cannot be changed. The negative .DAT slots are those used by the system and the user need not be concerned with them. The positive .DAT slots, however, are user .DAT slots. When writing programs which are to be run within the system, the user should be careful to use the correct .DAT slot numbers.

.DAT SLOT	DEVICE	HANDLER	USE
1	TTY Printer	(TTA.)	Teleprinter Output
2	TTY Keyboard	(TTA.)	Keyboard Input
3	Paper Tape Reader	(PRA.)	Input
4	TTY Printer	(TTA.)	Listing
5	Paper Tape Punch	(PPA.)	Output
6	Paper Tape Reader	(PRA.)	Scratch
7	Paper Tape Punch	(PPA.)	Scratch
10	Paper Tape Reader	(PRA.)	Scratch

For example, if the user desires to output to the teleprinter from a FORTRAN IV program, the WRITE statement should read:

WRITE (1, 10), where 1 is .DAT slot 1 and 10 is the FORMAT statement number.

APPENDIX 2

PDP-9 ASCII CHARACTER SET

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

	00-37	40-77	100-137	140-177	
	ASCII CHAR.	ASCII CHAR.	ASCII CHAR.	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

	00-37	40-77	100-137	140-177	
	ASCII CHAR.	ASCII CHAR.	ASCII CHAR.	ASCII CHAR.	
*33	ESC	;		ESC	33
34		<			34
35		=		ESC	35
36	RS ( )	>	∧ or ↑		36
37		?		delete (RO)	37

*Codes 33, 173, 175 are interpreted as ESC (ALT Mode) and are converted on input to code 175 by IOPS handlers.

APPENDIX 3

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 4

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 5

SYSTEM RESTART

SYSTEM RESTART can be used to attempt to restart a system program (excluding DDT and the oder) which has halted during operation. It is most easily used if the symbolic program (SYSTEM START), shown on the following page, has been punched onto tape and assembled by MACRO-9. If is has been done and a binary tape is available, proceed as follows:

1. place binary tape in reader
2. push tape feed button to clear end-of-tape flag
3. set address switches to 17720
4. press I/O RESET
5. press READIN

If a binary tape is not available, SYSTEM RESTART may be entered through the AC switches, ing the deposit key. When this has been done, proceed as follows:

1. set address switches to 17720
2. press I/O RESET
3. press START

If the restart attempt is successful, the effect will be the same as a control P restart. Further starts are possible without reloading SYSTEM RESTART by the following procedure.

1. set address switches to 17720
2. press I/O RESET
3. press START

If the restart attempt fails, it will be necessary to reload the system program tape.

Since a halt usually indicates a rather serious problem, SYSTEM RESTART will often fail to start the program.

Two instances in which it will work

1. When the user has accidentally typed an ALT mode instead of a carraige return as a command string terminator, and wishes to regain the program for another run.
2. After an IOPS 3 error if the offending device flag has been removed.

```
                                        .TITLE SYSTEM RESTART
                                        .FULL
17720                                   .LOC 17720
17720       707702          D           EEM
17721       237734                      LAC* E
17722       057731                      DAC  A
17723       703302                      CAF
17724       700042                      ION
17725       237732                      LAC* B
17726       705504                      ISA
17727       177733                      DZM* C
17730       637731                      JMP* A
17731       000000          A           0
17732       000006          B           6
17733       001413          C           1413
17734       000632          E           632
            017720                      .END D
                                        NO ERROR LINES
```

APPENDIX 6

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
30	API software level error	API status register

APPENDIX 7

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 shouldn't 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 non-common 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 7A
FORTRAN IV ERROR LIST

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

ror 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 non-zero 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 parenthesis.
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 non-format 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.

Error Code	Cause
040	Adjusted floating point exponent exceeds 76.
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 parenthesis.
060	Illegal mode mixing.
063	Signed assignment variable.
069	Doubly defined statment 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 shouldn't 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.

Error Code	Cause
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
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.

Error Code	Cause
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.
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.