The current release of the PDP-9 ADVANCED Software System (Keyboard Monitor V4B) supercedes all earlier versions and is distributed in one of two forms:

1. Three (3) trays of paper tapes (24 tapes) for PDP-9's with Disks but no DECtapes.

2. One (1) DECtape for PDP-9's with Disks and/or DECtapes.

DECTAPE PDP-9's

The System DECtape issued by Digital Equipment Corporation is an 8K, non-EAE, non-API, 33KSR version of the Keyboard Monitor System: V4B, 9-30-68 (DEC-9U-SRZB-UC). SGEN (System Generation) should be used to tailor the system tape to the specific machine configurations. A write-up on SGEN starts on page .

DECTAPE/DISK PDP-9's

The System DECtape issued by Digital Equipment Corporation is an 8K, non-EAE, non-API, 33KSR version of the Keyboard Monitor System: V4B, 9-30-68 (DEC-9U-SRZB-UC). Disk handlers are contained in the binary file, .LIBR BIN. It is not directly operable from the Disk until system generation has been performed. Before attempting to use the system, the user should be familiar with the material contained in the SGEN write-up beginning on page , and the PDP-9 ADVANCED SOFTWARE DISK SYSTEM: INTERIM VERSION write-up, beginning on page .

In addition to the System DECtape (DEC-9U-SRZB-UC) and DECtape bootstrap (DEC-9U-LBAA-PH), the following two paper tapes should be included:

1. .DSKBT (DEC-9U-LWDA-PH) - Disk System Bootstrap

2. DSKSAV (DEC-9A-UPFA-PH) - Disk System Save/Restore Utility Program

PAPER TAPE/DISK PDP-9'S

The 24 binary tapes, as distributed by Digital Equipment Corporation, constitute an 8K, non-EAE, non-API, 33KSR version of the Keyboard Monitor System: V4B, 9-30-68 (DEC-). SGEN (System Generation) should be used to tailor the system to the specific configuration involved. Before attempting to use the system, the user should be familiar with the material contained in the SGEN write-up, beginning on page , and the PDP-9 ADVANCED SOFTWARE DISK SYSTEM:

INTERIM VERSION write-up, beginning on page

In addition to the three trays of tape, the following two paper tapes should be included:

1. .DSKBT (DEC-9U-LWDA-PH) - Disk System Bootstrap

2. DSKPTR (DEC-9A-YPDA-PH) - Disk System Save/Restore Utility Program

Attached is a list of PDP-9 ADVANCED Software System changes between V4B, 9-30-68 and earlier versions, a write-up on the INTERIM DISK SYSTEM, one on MAGTAPE HANDLER: MTF., one on the RMØ9 drum handlers, and additions to the FORTRAN IV subroutines library. Also included are features published with system V3C which have not yet been up-dated in manuals.

Document Contents:

PART I

PDP-9 ADVANCED SOFTWARE SYSTEM CHANGE NOTICE
VERSION V4B (9-30-68)

I/O Handlers

A. Disk Handlers (DKA., DKB., DKC., DKD.)

Disk handlers (DKA., DKB., DKC., DKD.) have been added to the system library (see PDP-9 ADVANCED SOFTWARE DISK SYSTEM: INTERIM VERSION, page 30).

B. Magnetic Tape Handler (MTF.)

Magnetic tape handler MTF. has been added to the system library (see MTF., Version 1, page 44). When used in the Keyboard Monitor environment, MTF. references .SCOM+4, bit 6 to determine whether 7 or 9 channel drives are present in the system (Ø=7 channel, 1=9 channel). Variable MT9CHN need only be defined for 9-channel paper-tape systems.

C. Teletype Handler (TTA).

Appropriate delays for both vertical and horizontal tabbing have been added to the teletype handler to accommodate the automatic tabbing mechanisms of Model 35 teletypes.

D. Card Reader Handler (CDB.)

CDB. has been modified to operate in the following manner:

1. *End-of-Medium: EOM occurs on 1 of 2 conditions:

   a. The presence of an EOF card (col. 1 = all 1's).

      NOTE: An EOF card is required as the terminating card of a deck when using the CRØ1E reader. It is optional for the CRØ2B.

   b. Striking the EOF button (CRØ2B only) with the hopper empty. This feature is used as follows:

      (1) Hopper empty (at end of deck) causes teletype output of IOPS 4 (not ready).

*End-of-Medium on the card reader is equivalent to an End-of-File, i.e., the card reader handler recognizes this condition as the end of the entire card deck or file.

(2) At this point striking the EOF button on the card reader followed by striking ↑R on the teletype causes recognition of end of file.

NOTE: The IOT, CREF, 7Ø6761, skip on EOF, is recognized only by the CRØ2B control and is assumed to be an effective NOP on the CRØ1E.

2. Reader-Not-Ready (.IOPS 4): All error conditions result in .IOPS 4. Manual intervention at the card reader allows continuation of card input:

a. Hopper empty

b. Stacker full

c. Not ready: Stop button depressed
Start button <u>not</u> depressed
Validity check with validity button on

d. Feed check

e. Read check

E. <u>DECtape Handlers (DTA., DTB., DTD., DTC.)</u>

1. All the DECtape handlers have been modified to allow up to 56$_{10}$ files on a DECtape unit that is not a system device. (24$_{10}$ is still the maximum of the system device.) Block 1ØØ$_8$ is the directory; blocks 71-77 are now all reserved as file bit map blocks.

2. DTB. has been shortened to allow DECtape input and output for the assembler (MACROA) and the compiler (F4A) in 8K systems.

3. .MTAPE monitor calls to the DECtape handler DTD. are now operational. This is the only handler which recognizes them.

4. DUMP and IMAGE data modes are now operational in DTA. and DTD.

F. <u>Drum Handlers (DRA., DRB., DRC., DRD)</u>

1. Four drum (RMØ9) handlers have been added to the system library for use in the PDP-9 Keyboard Monitor environment (see RMØ9 Drum Handlers, page).

G. ALL I/O Handlers

The .WAITR 2, ADDR monitor call has been implemented in all the I/O device handlers.

In I/O busy situtations control is given to the location with the symbolic name ADR; otherwise it is identical to the .WAIT a monitor call.

KM9 V4B

A. NEW FEATURES

1. Channel(C) 7 (or 9)

This command causes the default operation bit (.SCOM+4, bit 6) to be cleared or set. If this bit is Ø, then 7 channel operation will be assumed by the magtape handler. If it is 1, then 9 channel will be assumed. This default condition can also be set at system generation time by answering YES or NO to the question -

"SHOULD DEFAULT ASSUMPTION BE 7 CHANNEL MAGTAPE?"

2. 339(3) ON/OFF

This command tells the monitor that a 3Ø$_8$ register block is (or is not) reserved for the 339 push-down list. If a 339 display handler is later loaded and the push-down list has not been reserved, an .IOPS 24 error will occur on the first .INIT to that handler. The default condition of the 339 ON/OFF bit (.SCOM+4, bit 5) can be set at system generation time by answering YES to the question -

"SHOULD DEFAULT ASSUMPTION BE A 339 LOAD?"

3. VC38(V) ON/OFF

This command causes the character display table for the VC38 option to be set up prior to loading of any system or user programs. If this table is not set up and the user does not specify a character table in an .INIT to the handler, the display handler will assume the absence of a VC38 for text manipulation. The default condition of the VC38 ON/OFF bit (.SCOM+4, bit 4) can be set at system generation time by answering YES to the question -

"SHOULD VC38 CHARACTER TABLE BE LOADED?"

If the 339 is ON (bit 5 of .SCOM+4=1), then the contents of the first location of the push-down list is initially set to zero. However, if VC38 is ON (bit 4 of .SCOM+4=1), then the first register of the push-down list points to the starting address of the VC38 table.

This command implies 339 ON.

4. BATCH CD (Reference Addendum Number 2 to PDP-9 Monitor, Manual DEC-9A-MABØ-D)

This command causes batch mode to be entered with the CRØ1E or CRØ2B as the batch input device. Two special characters may be keypunched to generate proper commands.

a. Back arrow (←) = 0-5-8 punch

b. Altmode (ESC) = 12-1-8 punch

5. Expansion of the SKIP BLOCK

The skip block has been divided into two separate parts:

a. SKPBLK (logical block 44) which contains the skip chain and associated transfer vectors.

b. IOBLK (logical block 55) which contains the I/O handler names as produced by SGEN.

6. F4A is present on the System Tape and may be called.

7. GET function:

In order to allow a SAVE (↑Q) and restart of all system programs and to compensate for all changes which result from Q occuring during execution of an I/O handler, particularly TTA, the GET function now works as follows:

a.) $GET n XXXXX↲
works as in the past where n = the unit # of the device whose save (↑Q) area is being referenced and XXXXX is the location in memory where execution is to begin.

b.) $GET n (H)ALT↲
when the save area is loaded the computer stops. The user may load the address switches with his desired starting location (remembering to raise the extended memory switch if the machine is greater than 8K) and press "START" or follow the directions in (c) below.

c.) $GET n
When the save area is loaded, execution of a Monitor wait loop with API and/or PI on tapes place.
To start the saved program, one of the following actions must take place:
(1) Type ↑P or ↑T or ↑S
or (2) Press the STOP button; set the desired start address in the address switches; and hit START (extend switch up if greater than 8K).

↑P restarts system programs and will restart any user program which has issued an .INIT to the teletype with a restart address.

↑T is for a DDT restart.

S would be used only if a SAVE (↑Q) had been executed at the completion of a link load when the loader was waiting for ↑S to be typed.

8. Communication with PIP:

For improved PIP operation (see PIP V7A), KM-9 has been changed to refrain from storing the unit # in the .DEV table.

9. A major improvement in Monitor operation with reference to IOPS errors has been implemented in V4B. At the occurence of all IOPS errors other than Ø4 (not ready - which operates as in the past), the resident Monitor now waits in a teletype loop for the user to type ↑P to restart the system or user program; ↑Q to execute a save of core; or ↑C to reload the Monitor.

10. Additional Features Which Were in KM9 V3C

a. New and/or Modified Commands

(1) INSTRUCT (I) List of monitor commands

(2) INSTRUCT (I) ERRORS List of .IOPS, .LOAD, .SYSLD, .OTS and .SEG error messages

All other INSTRUCT commands have been deleted since this information is contained in the Keyboard Monitor Guide (DEC-9A-NGBA-D).

(3) ASSIGN(A) Assumes handler A if a two-letter handler is typed and Unit Ø if no unit number number is typed.

Example: A DT1 2 is equivalent to A DTA1 2
and A DT 3 is equivalent to A DTØ 3

(4) BATCH(B) XX Enter batch rpocessing mode. XX can be PR (paper tape reader) or CD (card reader).

(5) PATCH Bring in system patcher.

(6) CHAIN Bring in chain builder.

(7) EXECUTE(E) NAME   Begin executing the chain file called NAME.

(8) MACROA   Bring in abbreviated MACRO-9.

(9) F4A   Bring in abbreviated FORTRAN IV.

(10) API ON/OFF   Run the system with the API on or off.

b. New Options

Resident error routines for
    Power Fail
    Memory Parity
    Non-existent memory violation and
    Memory Protect violation
have been incorporated into the system and effect an .IOPS error message if the condition is detected. (See .IOPS error list below.)

The user has the ability (via .SETUP) to tie his own handler into these skip IOT or API channel registers.

Example: A machine with power fail will generate an .IOPS 34 error if the power low flag interrupts. The IOPS error routine will not save the appropriate registers. Therefore, if the user wants the ability to restart, the save and restore routines should be contained within the user's program. The skip IOT for power fail must be connected to this routine by a call to .SETUP.

Negative skip facility is now allowed in the program interrupt skip chain.

c. Clock Handling

.TIMER monitor call has been modified to allow the user to indicate another interval at the completion of previous interval in the subroutine specified to .TIMER.

```
LOAD  AC  with desired interval  (2's complement)
DAC*   (7
LAC  RETURN                  /RESTORE LINK
RAL
LAC SAVEAC                   /RESTORE AC
CLON                         /TURN ON CLOCK
JMP*  RETURN                 /RETURN TO USER
```

8. New Error Lists for V4B

Monitor Errors - .IOPS

Ø	ILLEGAL FUNCTION CAL - CAL ADDRESS
1	CAL* ILLEGAL - CAL ADDRESS
2	.DAT SLOT ERROR - CAL ADDRESS
3	ILLEGAL INTERRUPT - INPUT/OUTPUT STATUS REGISTER
4	DEVICE NOT READY - TYPE R WHEN READY
5	ILLEGAL .SETUP CAL - CAL ADDRESS
6	ILLEGAL HANDLER FUNCTION
7	ILLEGAL DATA MODE - CAL ADDRESS
1Ø	FILE STILL ACTIVE - CAL ADDRESS
11	SEEK/ENTER NOT EXECUTED - CAL ADDRESS
12	UNRECOVERABLE DECTAPE ERROR - 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
2Ø	DISK FAILURE - DISK STATUS REGISTER
21	ILLEGAL DISK ADDRESS - CAL ADDRESS
22	TWO OUTPUT FILES ON ONE UNIT - CAL ADDRESS
23	ILLEGAL WORD PAIR COUNT - CAL ADDRESS
24	339 PUSH-DOWN LIST NOT RESERVED - CAL ADDRESS
25	ILLEGAL DRUM SECTOR ADDRESS
26	ILLEGAL DRUM UNIT
27	ILLEGAL DISK UNIT - CAL ADDRESS
3Ø	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
34	POWER FAIL SKIP NOT SETUP - PROGRAM COUNTER
35	ILLEGAL DRUM SIZE

Loader Errors - .LOAD or .SYSLD

1	MEMORY OVERFLOW
2	DATA ERROR
3	SUBROUTINE NOT FOUND
4	.DAT SLOT ERROR

Object Time System Errors - .OTS

5	ILLEGAL REAL SQUARE ROOT ARG
6	ILLEGAL DOUBLE SQUARE ARG
7	ILLEGAL INDEX IN COMPUTER GOTO
1Ø	ELLEGAL I/O DEVICE NUMBER
11	ELLEGAL INPUT DATA
12	ELLEGAL FORMAT STATEMENT
13	ILLEGAL REAL LOGARITHMIC ARG
14	ILLEGAL DOUBLE LOGARITHMIC ARG

Execute Errors - .SEG

1 CHAIN CANNOT CALL ITSELF
2 ILLEGAL EOF
3 ILLEGAL EOM
4 READ ERROR
5 CHAIN AND COMMON TOO LARGE

SYSTEM GENERATOR V3B

A. CORRECTED BUGS

1. Special device handlers can be specified with no skip IOT's in the skip chain.

2. Negative skip IOT'S - .SGEN has been corrected to allow placement of a negative skip (SKIP ON NOT CONDITION) anywhere in the skip chain.

B. NEW FEATURES

1. Magtape 7 or 9 channel default assumption can be set at system generation time if any magtape handlers were requested.

2. 339 system load and VC38 option default assumptions can be set at system generation if a 339 handler is added.

3. System devices are now DT or DK (DTC or DKC assumed).

C. USER'S HINT

1. The resident monitor can be reduced in size by deleting unwanted devices from the system. This can be accomplished by typing N for those devices (actually device handlers) which are not present in the user's configuration (e.g., Disk (DKA, etc.), Mattape (MTF, etc)).

2. Another way of deleting skip IOT's is available to those installations not having Memory Parity, Power Fail, or Memory Protect. When skip chain ordering is performed, type all skips except those associated with the options which are not included at the installation; namely -

   a. SPE for memory parity

   b. SPFAL for power fail

   c. MPSNE and MPSK for memory protect

   Then type DONE to terminate the skip chain.

3. Note to 24 and 32K PDP-9 Users

   Because of the expansion of the library file (.LIBR BIN) in the Keyboard Monitor system, system generation of 24 and 32K systems requires one of two possible preliminary steps to assure the existence of adequate space on the generated system tape:

   a.) The simpler of the two methods is to modify .SGEN, cell 13475 (NOP), thereby inhibiting the provision of a Q area on the generated tape. (This does not inhibit use of Q to other tapes).

b.) Prior to system generation, run the UPDATE program and delete unwanted I/O handlers from .LIBR BIN; e.g., drum, disk, etc. if these devices do not exist in the system.

Please note the write-up on system generation (page)

UPDATE V4A

A. CORRECTED BUG

1. The binary program processing routines have been modified to allow for short (1 binary buffer) programs.

B. ADDITIONAL FEATURES

1. The command

```
DELETE (D) NAME1, NAME2
```

will delete programs NAME1 through NAME2 inclusive and leave the file positioned immediately before the entry after NAME2.

DUMP V3A

A. ADDITIONAL FEATURES

1. The command NNN#-↲ or ALT Mode will dump DECtape block number NNN read in the reverse direction.

MACROA V3A

A. CORRECTED BUG

1. The buffers in MACROA have been lengthened to allow for card image (81 characters) input.

B. PERMANENT DESIGN RESTRICTIONS

1. User defined Macros are not allowed. (System Macro calls are still allowed)

2. .ABS and .FULL output is not allowed.

3. Conditionals and Repeats are not allowed.

C. SYMBOL TABLE

1. The following symbol table sizes exist on 8K with the full complement of skip IOT's in the skip chain:

   (NOTE: Handlers listed are for .DAT slots -11, -12, -13, and -10 respectively)

   a. PRB,TTA,PPC,TTA - 610 symbols (decimal)

   b. DTC,TTA,PPC,TTA - 482 symbols (decimal)

   c. DTB,TTA,DTB,TTA - 261 symbols (decimal)

F4 V4A

A. CORRECTED BUGS

1. Correct code is generated for a subroutine argument which is in both a DIMENSION and a COMMON statement.

2. References to non-existent memory: Several illegal references to non-existent memory by F4 and F4A have been found and corrected.

3. Absence of an END statement: The compiler has been modified to assume the presence of an END statement if it fails to find one. This corrects the endless error printout of former versions.

B. TEMPORARY RESTRICTIONS

1. EXTERNAL statements in subprograms will not compile properly.

2. Function names which are used as dummy arguments in subprograms will not compile properly.

3. The compiler fails when a subprogram contains a COMMON statement and a dimensioned array is in the argument list.

Programming hint to temporarily get around the bug:

NORMAL CODING

```
      Main program
           DIMENSION A(2ØØ)
           COMMON B

           CALL SUB (A)

      Subroutine
           SUBROUTINE SUB(A)
           DIMENTION A (2ØØ)
           COMMON B
```

CORRECTED CODING (with new code underlined)

```
      Main Program
           DIMENSION A(2ØØ)
           COMMON B

           CAL SUB (A(1))

      Subroutine
           SUBROUTINE SUB (AX)
           DIMENSION A(1)
           COMMON B
           CALL ADJ1 (A,AX)
```

At this point the array A can be referenced as it would be in the normal coding above.

4. Mixture of .OR. and .AND. in logical IF statements generate errors at compilation time. This can be corrected by breaking up the logical expression in statements prior to the logical IF statement.

5. If the name of a subroutine or function does not agree with the file name given to the program, an E error will occure in PASS 1 as a warning. It will have no effect on the compiled output.

6. Block data subprograms with COMMON blocks are not properly compiled.

C. <u>SYMBOL TABLE</u>

1. The following symbol table sizes exist on 8K with the full complement of skip IOTs in the skip chain:

   (NOTE: Handlers listed are for .DAT slots -11, -12, and -13 respectively)

   a. PRB,TTA,PPC - 189 symbols (decimal)

   b. DTC,TTA,PPC - 62 symbols (decimal)

D. <u>MANUAL CHANGE</u>

1. Changes need to be made to the PDP-9 FORTRAN IV Manual (DEC-9A-AF4B-D)

   a. Add the following paragraph to the description of the EQUIVALENCE statement of page I-37.

   An EQUIVALENCE statement which refers to an array which also appears in a COMMON statement must refer to the first element of that array.

```
Example:    DIMENSION      A(100), B(50)
            COMMON         A
            EQUIVALENCE   (A(1), B(1))
```

   The above example shows a legal EQUIVALENCE statement. If, however, the statement were changed to read EQUIVALENCE (A(50), B(1)), the statement would not compile.

   b. Add the following paragraph to the description of TYPE statements on page I-35.

   Any function reference (statement functions, intrinsic functions, or external functions) which are not implicitly REAL or INTEGER must appear in the appropriate type statement.

```
Example:    DOUBLE PRECISION B,X,DABS,DATAN

            B=DATAN(DABS(X))
```

   If DABS and DATAN were not declared DOUBLE PRECISION, improper code would be generated by the compiler.

F4A V4A (New)

A. PERMANENT DESIGN RESTRICTIONS

1. F4A has been deliberately designed for use on 8K installations. Installations with 16K or greater may always use F4. Consequently, F4A is not intended to operate in systems greater than 8K.

2. The O (object code listing) and S (symbol table listings) options have been deleted.

3. The following statements have been deleted from FORTRAN IV:

   a. EQUIVALENCE

   b. EXTERNAL

   c. ASSIGN

   d. Assigned GOTO

B. TEMPORARY RESTRICTIONS

1. All which apply to F4 apply to F4A.

C. SYMBOL TABLE

1. The following symbol table sizes exist on 8K with the full complement of skip IOTs in the skip chain:

   (NOTE: Handlers listed are for .DAT slots -11, -12, and -13 respectively.)

   a. PRB,TTA,PPC - 407 symbols (decimal)

   b. DTC,TTA,PPC - 278 symbols (decimal)

   c. DTB,TTA,DTB - 54 symbols (decimal)

FORTRAN IV SUBROUTINE LIBRARY

A. CORRECTED BUG

The double precision multiply routines for EAE have been improved from 33 to 34 bit accuracy, with the 35th bit rounded off.

LINKING LOADER V3A

A. ADDITIONAL FEATURES

1. The Linking Loader has been changed to allow common blocks and blockdata subprograms to overlap memory banks. FORTRAN IV allows common block sizes greater than 8192_{10}, provided that each element in common does not exceed 8192_{10}, e.g.,

   ```
   COMMON  /I/L(100,100)
   ```

   is illegal because the size of array L is 10000_{10}.

   ```
   COMMON  /I/L1(100,50),L2(100,50)
   ```

   is acceptable. Each array size is 5000_{10} and the size of the common block is 10000_{10}.

   (NOTE: FORTRAN IV does not properly compile blockdata subprograms with common blocks.)

2. Rubouts (echoes\) and CTRL U (@) may be used in the command string to delete characters or the entire line. Carriage return is equivalent to comma except that it allows continuation of the command string on the next line. ALT Mode terminates the command string.

3. CTRL P (↑P) may be used to restart the Loader if an error is made in typing the command string. CTRL P (↑P) must be given before the terminating ALT Mode to be effective.

4. Non-common arrays and variables are initialized to zero (0) by the Loader.

5. MACRO-9 programs can be linked to COMMON areas defined by FORTRAN IV. If any virtual globals remain after the Loader has searched the user and system libraries and defined COMMON blocks, an attempt will be made to match those global names to COMMON block names. If a match is made, the global becomes defined as the address of the COMMON block. An example follows:

   FORTRAN IV PROGRAM

   ```
    INTEGER A,B,C
    COMMON /NAME/C
    COMMON A,B
          .
          .
          .
   ```

MACRO-9 PROGRAM

```
 .GLOBL NAME, .XX    /.XX is name given to blank COMMON
                     /by the F4 Compiler
 DZM*   .XX          /CLEAR A - NOTE INDIRECT REFERENCE
 ISZ    .XX          /BUMP COUNTER
DZM*    .XX          /CLEAR B
DZM*    NAME         /CLEAR C
```

Please note that if the values are REAL (2 words) or DOUBLE PRECISION (3 words), the MACRO program must account for the number of words when accessing specific variables.

EDITOR V4A

A. CORRECTED BUGS : BLOCK MODE IMPROVEMENTS

1. Block Buffer is now properly cleared on program restart.

2. Final line of buffer may now be deleted.

3. Buffer emptied by deletions may now be filled by READ request.

4. WRITE request with pointer in middle of buffer will output entire buffer.

5. READ request no longer required before teletype input to prime buffer.

6. WRITE request no longer required before CLOSE.

7. RENEW command acts just like WRITE followed by READ.

8. BLOCK OFF issued while buffer is partially full results in message:

   BUFFER NON-EMPTY

   User must empty buffer before block mode really goes off. Editor cannot just blindly write buffer because pointer position would get lost in the process.

B. ADDITIONAL FEATURES

1. Temporary File Deletion

   If the Editor's temporary file (.TFIL1 EDT) is found on the output device, the following comments are printed:

   FILE .TFIL1 IS PRESENT ON OUTPUT DEVICE. PLEASE RENAME IT OR IT WILL BE DELETED.

   If the file is not explicitly deleted or renamed (via the CALL function), it is automatically deleted.

2. Like-Named Files

   a. If a file is found at TOP or CLOSE time in the "output" device, the following comments are printed:

   FILE FILNAM EXT IS PRESENT ON OUTPUT DEVICE.

   YOUR EDITED FILE IS ON .DAT$\begin{Bmatrix}-14\\ \\-15\end{Bmatrix}$ AS $\begin{Bmatrix}\text{.TFIL1 EDT}\\ \\ \text{FILNAM EXT}\end{Bmatrix}$.

ORIGINAL FILE DELETED.

The file processing goes as follows:

1. The Editor first attempts to leave the file on -14 as FILNAM EXT.

2. If it cannot do (1), it tries to leave the file on -14 as .TFIL1 EDT.

3. If it cannot do (2) either, it leaves the file on -15 as .TFIL1 EDT.

4. The Editor is restarted.

   In any case, the user knows exactly the status (residence, name) of the edited file.

   This scheme protects against:

   (1) Files of the same name on -14 and -15.

   (2) Two or more files of the same name on -14.

3. Name Provided in CLOSE sequence

It is now possible to give a name in the CLOSE request which is identical to that given in the OPEN sequence. The normal comment:

    PLEASE USE ANOTHER NAME

is not printed in this case.

4. Provision for Backup Files

The KEEP function effects preservation of the original file as it appears on .DAT -14. That file will be renamed "SAVNAM EXT" during processing of the next CLOSE or TOP request.

Format:

    KEEP SAVNAM (EXT)

"SAVNAM" must be different from any file name appearing in the directory on -14. As many KEEP requests as needed may be issued.

Restriction:

1. The KEEP request must be issued while the input file is on -14 (i.e., the number of TOP's issued must be even 0,2,4,...).

2. "SAVNAM EXT" must be unique. If it is not, the name "SAVFIL EDT" will be assigned.

Example of Usage:

```
    EDITOR V4A
    OPEN THSFIL
    EDIT
    KEEP BKUP01

    TOP                   /Normal TOP
    NEXT                  /Some locative request must
                           be issued here to allow
                           physical file transfer at
                           second TOP.
    TOP                   /To return file to -14.
    KEEP BKUP02

    CLOSE
```

Resulting Directory on -14:

```
    THSFIL SRC            /New edited file
    BKUP01 SRC            /First copy--original
    BKUP02 SRC            /Second copy--after TOP's
```

5. File Renaming and Deletion

The CALL function allows user access to the file renaming and deletion facilities in the Editor. This command may not be issued while any file (either input or output) is open. That is, it may only be used immediately after the Editor is loaded or restarted.

Format:

```
                INPUT
CALL RENAME              FILE1 EX1 FILE2(EX2)
                OUTPUT
```

- The file FILE1 EX1 on .DAT-14 (if INPUT is specified) or .DAT-15 (if OUTPUT is specified) is given the new name FILE2. EX2 need not be given. If it is not, EX1 is used; if it is given, then EX2 is used.

```
                INPUT
CALL DELETE              FILNAM EXT
                OUTPUT
```

The file named FILNAM EXT is deleted from the device specified (INPUT or OUTPUT). EXT need not be given and, if it is not, "SRC" is used.

6. ICLOSE Request

The ICLOSE command effects the closing of the current input file only. The output file remains open. A new input file may be referenced after the ICLOSE request by issuing an OPEN command. ICLOSE provides a facility for combining source files during an editing run.

C. PRECAUTIONS

1. Block Buffer Overflow

In the event that the editor's block buffer becomes full (BUFFER CAPACITY EXCEEDED BY: Offending line) the user's only recourse at present is deletion of the line (or lines) that caused the error. The user must ensure that the buffer contents stay well within the buffer limits if he is to avoid this situation.

2. SCLOSE Restriction

The command SCLOSE is intended to allow the placement of the edited file on the current output device. Its use, in place of the CLOSE request, normally causes the file to be left on .DAT -15.

This allows the closing of long files, after trivial changes, without the tedious recopy process from -14 to -15. However, its use causes a fatal error under some circumstances. For example, the following sequence causes an IOPS 1Ø error:

```
>ICLOSE

>SCLOSE NAME                   where name is the same as
 Error message (Please:        that used in the OPEN command
 Use another name)

>SCLOSE NEWNAM
```

For this reason, SCLOSE should be used only with care.

DDT V3A

A. ADDITIONAL FEATURES

1. DDT can now be called without loading any user programs. When the Linking Loader is in core and types

   LOADER V2A

   waiting for the user to type in the names of the programs to be loaded, the user may type CTRL T (↑T). Control will go directly to DDT.

2. The HDR$ routine (which changes the symbol table look-up) has been modified. The proper form is now

   NAME;HDR$

   with no spaces.

   The old form (NAME HDR$) will still work if NAME is not also used as an internal symbol.

B. DELETIONS

1. The patch file feature has been deleted from the Keyboard Monitor version of DDT. Thus, the following commands are illegal and will produce a question mark (?) response:

   PFI$

   PFO$

   SNS$

   PFE$

   .DAT slots -6 and -10 have no further significance to DDT.

C. RESTRICTIONS

1. Breakpoints should not be placed in routines which operate with API active. If such a breakpoint is encountered, DDT will type:

   API 4XXXXX   (API status register)

   along with the normal breakpoint information. The exclamation point (!) will have no effect at that breakpoint, that is, a restart from the breakpoint is not possible.

PIP V7A

A. CORRECTED BUGS

1. An error is V5A which prevented H mode copy of a disk unit to DECtape & vice versa has been corrected in V6A.

2. An error in V5A which caused failure to copy block 1077 in H mode copy of DECtape to DECtape has been corrected.

3. ASCII line buffer sizes have been increased to accomodate the 120 character line printer length.

4. H Mode Copy Problem in 24 and 32K: PIP has been corrected to properly copy tapes in H mode without false reports pf read-compare errors in 24 and 32K.

5. File Closing Problem: PIP has been corrected to close files after copy (H mode) operations to prevent the IOPS 17 (too many files referenced) after several copy operations, each referencing different .DAT slots.

B. ADDITIONAL FEATURES

1. Transfer, Copy, and List Directory commands may be used with 56 file DECtapes.

   WARNING:

   24 file tapes with earlier versions of the PDP-9 Advanced Software may still be used, provided they contain no more than 24 files. To convert to the new 56 file format, use the PIP Copy command with switch N or S. Do not use the H switch.

   To distinguish 24 file tapes from 56 file tapes:

   A 24 file tape indicates 4 system blocks when the directory is listed.

   A 56 file tape indicates 10 or more.

2. The New Directory (N) command or switch will now cause 8 blocks to be reserved as file bit map blocks (required to implement the DECtape 56 file capacity). These blocks are: the directory (block 100_8) and 7 file bit map blocks (71_8 through 77_8).

3. S switch: as a function of installation core size, (determined at system generation time), the S switch refreshes a directory, reserving ↑Q area blocks as well as basic system blocks for directory and file bit maps. For example, in an 8K system, 32_{10} additional blocks (101_8 - 140_8) are specified as occupied in the directory (block 100_8).

S is now a permissible switch (the only one) in a Newdir (N) command. Whereas,

```
        N  DT2 ↵
```

reserves only directory and file bit map blocks,

```
        N  DT2 (S) ↵
```

additionally reserves ↑Q area blocks. S remains permissible wherever else it is specified in the PIP manual. However, S and N are mutually exclusive when they both are used as switches.

(NOTE: S no longer copies a system tape.)

4. I/O device designations in the PIP9 command string may now include a third letter, e.g., the letter "A" in the statement:

```
    L TTA ← DTB1
```

   is acceptable to PIP-9.

5. File name extensions SRC and BIN may be omitted in the Transfer (T) or Verify (V) commands if the appropriate data mode switch (A or B) appears in the command string, e.g.,

```
    T DT2 (A) ← DT1 FILE ↵
```

   effects a transfer of FILE SRC to DT2. (If program FILE (no extension) exists on DT1, it will be transferred)

```
    V DT1 FILE (A) ↵
```

   causes <u>FILE SRC</u> to be verified.

   <u>Conversely</u>: if SRC or BIN appears as an extension in the command string (T or V), data modes A or B (whichever applies) will be assumed, e.g.,

```
    T DT2 ← DT1 FILE SRC ↵
```

   is the same as T DT2 (A) ← DT1 FILE SRC ↵
   is the same as T DT2 (A) ← DT1 FILE ↵

6. F and T are two new switch options. They may only be used with the transfer command (T) and data mode A.

   The T switch causes the deletion of trailing spaces in an ASCII file. T, E, and C are mutually exclusive switches. The F switch causes the insertion of a form feed (FF) and carriage return (CR) after every .EJECT or 56_{10} lines if no FF has been thus far encountered.

7. A new switch (Q) has been added to delete sequence numbers from ASCII input lines during the Transfer command. The switch is primarily intended for card conversion to some

other medium. Columns 73-80 are deleted leaving a 72 character line with a carriage return as the 73rd character.

The Q switch is legal only with A data mode in a Transfer command. It may be combined with C (spaces to tabs) or T (delete trailing spaces).

8. The internal logic of the Copy (C) command with H switch specified (i.e., a direct DECtape copy in its entirety) has been changed to use all of available core memory for a very much faster copy (approximately 5 minutes in 8K and 3.5 minutes in 16K). The <u>fastest</u> possible copy is executed by assigning "NONE" to all positive .DAT slots except those needed for the DECtape prior to loading PIP for an H copy.

9. Use of Positive .DAT slots: PIP V7A has been improved in its method of referencing .DAT slot assignments:

    a.) PIP now used .DAT slot-3 (alsways TTA) when teletype I/O is requested thus obviating the necessity of TTA assignment to the positive side of the .DAT table and freeing up one more slot for another device.

    b.) PIP no longer requires the exact unit # of a multi-unit device in the positive .DAT table. Provided a handler for the device is assigned, PIP will set the unit # from the command string into the .DAT table.

    For example, suppose two slots were assigned to DECtape, units 1 and 2. PIP V7A will allow reference of any two DECtape units, not just 1 and 2 as in past versions.

PATCH V3A

(Reference Addendum Number 5 to PDP-9 Utility Manual DEC-9A-GUAB-D)

A. CORRECTED BUGS

1. Leading spaces are allowed in all commands and no longer cause an error.

2. Comments in some commands were flagged as errors. This has been corrected.

   Because of the way the READ command has now been implemented, a comment can only be used after both filename and extension have been given.

B. ADDITIONAL FEATURES

1. Single-Block Commands

   Forms:

   ```
   >B octnum↵
   >B+octnum↵
   >B-octnum↵
   ```

   The octal number specifies the block number: $\emptyset \leq \text{octnum} \leq 11\emptyset1_8$. If the number is out of this range, the following error message will be given:

   ILLEGAL BLOCK #

   B and B+ are equivalent and indicate that the block is to be read in the forward direction. B- means read in reverse direction. This distinction is necessary only for DECtape, which has bidirectional recording. For the Disk all three forms are equivalent.

   The Block commands are equivalent to specifying a system program for patching. Once a block has been selected, List and Read commands may refer to locations

   $$\emptyset \leq \text{loc} \leq 377_8$$

   Any other number causes ADDRESS OUT OF RANGE error. A space and optional comment may follow the block number in the Block commands.

2. The READ command has been altered to allow DECtape or Disk as the auxiliary input device (in addition to the paper tape reader). This was done because MACRO can now assemble Absolute Block Binary (.ABS) onto DECtape or Disk.

Forms:

```
>READ↵
>READ fname↵
>READ fname ext↵
```

The first form is used when reading from paper tape (no file name needed). If the second form is used, the extension ABS is assumed. Only the first 6 characters in fname are used and only the first 3 characters in ext are used. Each may contain up to 9 characters.

If the file is not found on the input device, the following error message may be given:

```
FILE NOT FOUND
```

3. When in single-block mode, if a READ command is given, the address range is from Ø to 377 octal and must be as such on the auxiliary device.

4. The error message

```
END OF MEDIUM
```

can no longer occur because end of medium cannot be detected anymore. This is due to the change in the READ command which now reads in dump mode.

CHAIN V2A

(Reference Addendum Number 6 to PDP-9 Utility Manual DEC-9A-GUAB-D)

A. CORRECTED BUGS

1. Chain numbers larger than 9 are now interpreted properly.

B. ADDITIONAL FEATURES

1. Common blocks and blockdata subprograms can now overlap memory banks (see Linking Loader V3A).

2. MACRO programs may refer to FORTRAN program common blocks through the use of global symbols (see write-up on Linking Loader V2A). This feature will work only if the MACRO and FORTRAN programs are part of the same chain. The only exception is blank common (.XX), which cannot be so referenced.

MONITOR V4B SYSTEM PROGRAM PATCHES

KM-9 V4B:

6234	4Ø772Ø	41372Ø	(New Values)
16312	1321	25Ø	
11122	64Ø14Ø	64Ø24Ø	
11123	554Ø67	554Ø71	
11124	576361	57636Ø	
13257	233537	75ØØØØ	
326	74ØØØØ	6ØØ434	
426	76ØØ56	76Ø111	
427	76Ø111	76Ø117	
43Ø	76Ø117	76Ø12Ø	
431	76Ø12Ø	76Ø123	
432	76Ø123	2ØØ24Ø	
433	76ØØ4Ø	62Ø422	
434	2ØØ24Ø	2ØØ511	
435	62Ø422	6ØØ377	
51Ø	4ØØ511	62Ø24Ø	
511	62Ø24Ø	25Ø	
5316	4Ø344	6Ø6413	
6413	Ø51Ø34	4Ø344	
6414	11Ø262	4Ø326	
6415	553466	6Ø5317	

.SGEN V3B:

145ØØ	316Ø21	316Ø41
13744	6ØØ11	6137Ø7

.SYSLD:

17363	3642	36Ø2
17364	764ØØ5	763745
17365	3643	36Ø3
17444	5Ø57	5Ø17
17445	765222	765162
17446	5Ø6Ø	5Ø2Ø
17353	7777	7737
17354	77Ø142	77Ø1Ø2
17355	1ØØØØ	774Ø

PATCH V3B:

16424	340432	341Ø32
16567	3642	36Ø2
1657Ø	764005	763745
166Ø3	5057	5Ø17
166Ø4	765222	765162
16561	7777	7737
16562	770142	77Ø1Ø2

Additions to the PDP-9 FORTRAN IV Subroutine Library

FILE

File Commands

1. Class: External Subroutine

2. Purpose: To provide the device-independent .IOPS commands SEEK, ENTER, CLOSE, FSTAT, RENAM, and DLETE. These commands are used to allow the FORTRAN IV Object Time System to communicate with .IOPS file-oriented devices.

   a. SEEK finds and opens a named input file.

   b. ENTER initiates and opens a named output file.

   c. CLOSE terminates an input or an output file and must be used if SEEK or ENTER has been used.

   d. FSTAT checks for the presence of a named file.

   e. RENAM checks for the presence of a file and renames it if found.

   f. DLETE checks for the presence of a file and deletes it if found.

   (NOTE: BACKSPACE, REWIND, and ENDFILE commands should never be used with a device that is operating in the file-oriented mode using the above subroutines.)

3. Calling Sequences:

   a. To seek a named file:

```
CALL SEEK (N,A)
```

where N=device number
      A=array name containing the 9-character 5/7 ASCII file name and extension.

The file array has the following format for the named file FILNAM EXT:

```
DIMENSION FILEN (2)
DATA FILEN(1),FILEN(2)/5HFILNA,4HMEXT/
```

To use this named file for input on .DAT slot 1:

```
CALL SEEK (1,FILEN)
```

b. To enter a named file:

CALL ENTER (N,A)

where N and A are the same as for SEEK.

c. To close a named file:

CALL CLOSE (N)

where N is the same as for SEEK.

d. To check for the presence of a named file:

CALL FSTAT (N, A, I)

where N and A are the same as for SEEK and I=-1 (.FALSE.) if file not found and I=0 (.TRUE.) if file found and action complete.

e. To rename a file A and call it B:

CALL RENAM (N, A, B, I)

where N, A(B is the same as A), and I are the same as for FSTAT.

f. To delete a named file:

CALL DLETE (N, A, I)

where N, A, and I are the same as for FSTAT.

4. External Calls:

FIOPS, .DA, .SS, .SEEK, .ENTER, .CLOSE, .FSTAT, .RENAM, .DLETE

5. Size: 322 octal locations

6. Error Conditions:

.OTS Error 10 if I/O device number is illegal
.IOPS Error 13 if file not found on SEEK
.IOPS Error 14 if directory full on ENTER

TIME — Clock Handling

1. Class: External Subroutine

2. Purpose: To provide the ability to record elapsed time in minutes and seconds

3. Calling sequence:

```
CALL TIME (IMIN, ISEC, IOFF)
```

This call causes the clock to be started and the elapsed time recorded as minutes and seconds in IMIN and ISEC. To stop the clock, set IOFF to non-zero.

Only one call to TIME or TIME1Ø can be active at any point in the user program.

Example:

```
 CALL TIME  (IM, IS, IOF)
 ⎧
A⎨
 ⎩
 IOF=1
 WRITE (4,1ØØ) IM, IS
```

This sequence will cause the time taken to execute the code at A to be output.

4. External Calls:

.DA, .TIMER

5. Size: 52 octal locations

6. Error Conditions:

None

TIME1Ø	Clock Handling

1. Class: External Subroutine

2. Purpose: To provide the ability to record elapsed time in minutes, seconds, and tenths of seconds.

3. Calling Sequence:

   CALL TIME1Ø (IMIN, ISEC, ISEC1Ø, IOFF)

   This call causes the clock to be started and the elapsed time to be recorded as minutes, seconds, and tenths of seconds in IMIN, ISEC, and ISEC1Ø respectively. To stop the clock, set IOFF to non-zero. Only one call at TIME1Ø or TIME can be active at any point in the user program.

   Example: See TIME

4. External Calls:

   .DA, .TIMER

5. Size: 66 octal locations

6. Error conditions:

   None

ADJ1 Adjustable Dimension

1. Class: External Subroutine

2. Purpose: To provide dimension adjustment on a one dimension array

3. Calling Sequence:

```
DIMENSION B(1)
    ≀
CALL ADJ1  (B,A)
```

where B is the array whose storage begins at A. A must be array element (such as C(2ØØ)) with sufficient storage beyond A to allow for all the entries of array B. The dimensions or type of array A do not have to agree with array B.

B cannot be a dummy argument in a subroutine but A can be a dummy argument.

Example:

```
DIMENSION A(3ØØ), B(1), C(1)
    ≀
CALL ADJ1 (B,A(1Ø1))
CALL ADJ1 (C,A(2Ø1))
    ≀
```

After the calls to ADJ1, the arrays B and C may be referenced as if they had been dimensioned as (1ØØ each. No further calls to ADJ1 have to be made.

4. External calls:

.DA

5. Size: 17 octal locations

6. Error conditions:

None

ADJ2 Adjustable Dimensionin

1. Class: External Subroutine

2. Purpose: To provide dimension adjustment for a two dimensio array

3. Calling Sequence:

```
DIMENSION B(1,1)
   ⋮
CALL ADJ2 (B,A,NR)
```

where NR is the number of rows to appear in array See ADJ1 for comments on B and A.

Example:

```
DIMENSION A(3ØØ), B(1,1), C(1,1)
   ⋮
CALL ADJ2(B,A(1),1Ø)
CALL ADJ2(C,A(1Ø1),2Ø)
   ⋮
```

After the calls to ADJ2, the arrays B and C may be referenced as if they had been dimensioned (1Ø,1Ø) and (2Ø,1Ø) respectively. No further calls to ADJ have to be made.

4. External Calls:

DA, .AD

5. Size: 36 octal locations

6. Error conditions:

None

ADJ3 Adjustable Dimensioning

1. Class: External Subroutine

2. Purpose: To provide dimension adjustment for a three dimension array

3. Calling Sequence:

```
DIMENSION  B(1,1,1)
CALL ADJ3  (B,A,MR,NC)
```

Where NR and NC are the number of rows and columns respectively to appear in array B. See ADJ1 for comments on B and A.

Example: See ADJ1 and ADJ2

4. External Calls: .DA, .AD

5. Size: 41 octal locations

6. Error Conditions:

None