VOLUME 3: #5

SOFTWARE PERFORMANCE SUMMARY

FOR THE

PDP-8 & PDP-12

SOFTWARE INFORMATION SERVICE

SOFTWARE PERFORMANCE SUMMARY

This Software Performance Summary is intended to supplement your set of Family of 8 software. It contains all current information on known software problems, patches, manual corrections and programming notes. Each article is coded sequentially by system program in the lower right hand corner. As new versions correct software problems and reprinted manuals include programming notes and manual corrections, new articles will announce the revised software and specify by this code which articles should be removed from your Software Performance Summary file. Articles may also be replaced when new information becomes available; such as, a procedure to circumvent a problem may replace the original report of the problem. This information, as well as newly discovered problems, methods, etc., is printed in Digital Software News for the PDP-8 & PDP-12. To assure that the monthly Digital Software News is sent to the appropriate software contact at your installation, please check with the Software Specialist or Sales Engineer at your local Digital Office.

Additional copies of the current Software Performance Summary and updated copies of the PDP-8/12 Software Price List are available at no charge upon request from the Program Library. As with usual orders for software, the Program Library will accept orders received directly from U.S. customers. All other customer orders must be routed through the nearest Digital Representative.

TABLE OF CONTENTS

PDP-8	KE8E EAE INSTRUCTION TEST 1, MAINDEC-8E-DOLA, Patch	MAINDEC	1
PDP-8	KE8E EAE INSTRUCTION TEST 1, MAINDEC-8E-DOLA, Patch	MAINDEC	2
PDP-8E	TELETYPE AND KL8 ASYNCHRONOUS DATA CONTROL TESTS, MAINDEC-8E-D2AB, Patch	MAINDEC	3
PDP-8	MI8-E BOOTSTRAP DIAGNOSTIC MAINDEC-8E-D1IA, Patch	MAINDEC	4
PDP-8	MR8-EA READ ONLY MEMORY TEST MAINDEC-8E-D1JA	MAINDEC	5
PDP-8	DF32 DISCLESS LOGIC TEST, MINIDISC MAINDEC-08-D5BB, Patch	MAINDEC	6
PDP-8	AD8E, AM8E A-D CONVERTER & MULTIPLEXER DIAGNOSTIC MAINDEC-8E-D6BB	MAINDEC	7
PDP-12	FLOATING POINT INSTRUCTION TEST 2C MAINDEC-12-DØOB, Patch	MAINDEC	8
PDP-8	PDP-8/E TELETYPE AND KL8 ASYNCHRONOUS DATA CONTROL TESTS, MAINDEC-8E-D2AC, Patch	MAINDEC	9
PDP-8	RF08 DISK DATA MAINDEC-08-D5EB, Patch	MAINDEC	10
PDP-12	MISCELLANEOUS	PDP-12 MISCELLANEOUS	1
PDP-12	PROGRAMMING NOTE Software Solution to a DJR Problem	MISCELLANEOUS	2
PDP-12	MASH USER'S MANUAL Additional notes on Automatic Mode	12 DOCUMENTATION	3
PDP-8	MISCELLANEOUS	PDP-8 MISCELLANEOUS	1
PDP-8/12	TC-58 HARDWARE INFORMATION	PDP 8/12 MISCELLANEOUS	1
PDP-8	RTPS FORTRAN IV (FORRTS.FT Version 4) Patch to correct logical input field	RTPS FORTRAN	1
PDP-8	RTPS FORTRAN IV (DEC-08-LRTDA) (F4.s. V.1.06) Problem with arithmetic functions	RTSP FORTRAN	2

AIPOS MONITOR

Problem on RUBOUT

In the AIPOS Monitor, on RUBOUT, the command interpreter sometimes stores information in the wrong file descriptor block. To correct this problem, the following binary patch should be implemented.

Block 1 (add 1 to the starting block) of JOB CONTROL

LOCATION	OLD CONTENTS	NEW CONTENTS
224	7ØØ1	7ØØØ

To implement a source patch, CMØ2, line 417 [IAC] should be deleted. The segment of the source should now read:

```
TAD I P1FAUX
AND P7
IAC
DCA FILCNT
JMP CMI18Ø
```

PDP-8 Binary Loader (DEC-Ø8-LBAA-PM)

Procedures to restore destroyed instructions.

The current version of the Binary Loader (DEC-Ø8-LBAA) dated May 10, 1967 is not compatible with either the Bootstrap Loader for the TCØ1 DECtape Library System (DEC-Ø8-LUAA) or with any program using the DISK Data Break (Locations 7750 & 7751).

The Binary Loader uses locations 7614-7616 as temporary storage of variables thus destroying the three instructions of the TCØ1 Bootstrap which should be in these locations. To restore the Bootstrap after reading in or using the Binary Loader the user should restore the contents of locations 7614-7616 as follows:

```
7614    6766    DCTA    DTXA
7615    3354    DCA     WCOUNT
7616    6771    DTSF
```

The TCØ1 bootstrap may then be started as usual at location 76ØØ. This version of the Binary Loader does not harm the Bootstrap Loader for the 552/555 DECtape Library System (Digital-8-3-U).

The Binary Loader also uses locations 7750 and 7751 for instructions. These two locations are used by the DF32 Mini Disk as its Word Count and Current Address registers for 3 cycle Data Break I/O transfers. Any disk I/O the user may do will, therefore, destroy the two instructions of the Binary Loader contained in locations 7750 and 7751. To restore the Binary Loader, restore the contents of these locations as follows:

```
7750    1355    TAD     WORD2
7751    5743    JMP     I ASSEMB .
```

The Binary Loader may then be started at 7777 and used as usual.

PDP-8 BINARY PUNCH PROGRAM (DEC-08-YXYA)

Failure to punch location 7777

The current version of the Binary Punch Program (DEC-08-YX1A-PB, Low Speed Punch; DEC-08-YX2A-PB, High Speed Punch) fails to punch out location 7777 when it is specified as the last location in a block. The following patch will correct this problem:

LOCATION	OLD CONTENTS	NEW CONTENTS
751Ø	7Ø41	7141
7515	71ØØ	237Ø
7516	237Ø	71ØØ

DISK DDT (DEC-D8-CDE1-PB and DEC-D8-CDE2-PB)

Patch to use high speed reader for reading symbol table tapes.

The patch given below will be of use to users who have no low speed paper tape reader and who wish to use Disk DDT but who are hampered by the fact that only the low speed reader may be used to read in user symbol table tapes. The patch is used as follows:

1) Load the program to be debugged into core.

2) Call DDT into core by typing DDT↵ in response to a Monitor "."

3) Stop DDT by depressing STOP on the console; then load the patch into core using the Binary Loader (system LOADER will destroy DDT). Note that the patch overlays one location in DDT (loc 3374) and thus the patch must be read in after DDT is in core.

4) Place the symbol table tape in the high speed reader with leader code under the read head.

5) Start the patch at location 4600[1]. The high speed reader will be initialized and control will pass to DDT, which will be expecting a command.

6) Type ALT MODE (or ESCape) R (echoes as [R) - the symbol table tape will read in.

7) After the last symbol has been read, DDT will type the new lower limit of the external symbol table, followed by an up-arrow (↑) to show that it is waiting for CONTROL P (↑P).

8) Type CONTROL P (↑P) - DDT will now be ready to accept commands, and the user may begin debugging, using the symbols he has just read in.

1 Users may change the location of the patch by changing the *4600 in the source and reassembling, being careful not to overlay any locations in his own program or any locations in DDT other than loc 3374, which must contain the constant READHS.

```
                        /ROUTINE TO ALLOW DISK DDT TO READ
                        /SYMBOL TABLE TAPE FROM HIGH SPEED
                        /READER
                        /
                        COMM=2513
                        START2-24Ø3
                        COMM34=3374
                        PUNOUT=3455
                        READKB=34Ø1
                        *COMM34
3374    46Ø3            READHS          /CHANGE DDT SYM TAB READ ROUTINE ADDR
              *46ØØ
46ØØ    6Ø14            RFC             /INITIALIZE HS READER
46Ø1    56Ø2            JMP 1  .+1      /JUMP TO
46Ø2    24Ø3            START2          /START2 IN DDT
46Ø3    ØØØØ    READHS,Ø
46Ø4    72ØØ            CLA
46Ø5    1244            TAD COMM1A      /GET ADDR OF NEW TABLE READ ROUTINE
46Ø6    3645            DCA I INCOMM    /STORE IT IN DDT
46Ø7    6Ø11            RSF             /WAIT ON FLAG
461Ø    52Ø7            JMP  .-1
4611    72ØØ            CLA
4612    6Ø16            RRB RFC         /GET CHARACTER
4613    3251            DCA TFM1        /STORE IT
4614    1251            TAD TFM1
4615    125Ø            TAD MIN2ØØ      /IS IT LEADER-TRAILER (2ØØ)
4616    745Ø            SNA
4617    52Ø4            JMP READHS+1    /YES - IGNORE IT
462Ø    1252            TAD CON1        /IS IT LINE FEED (212)
4621    765Ø            SNA CLA
4622    5234            JMP EXITA       /YES - GET CHAR AND EXIT
4623    1251            TAD TFM1
4624    1253            TAD CON1+1      /IS IT ROBOUT (377)
4625    745Ø            SNA
4626    52Ø4            JMP READHS+1    /YES - IGNORE IT
4627    1254            TAD CON1+2      /IS IT EOT (2Ø4)
463Ø    765Ø            SNA CLA
4631    5236            JMP FIXUP       /YES - RESTORE CHANGED DDT LOCS
4632    1251    RETN,   TAD TEM1        /NO - GET CHAR
4633    4647            JMS I OUTPUN    /ECHO IT
4634    1251    EXITA,  TAD TFM1        /EXIT WITH CHAR IN AC
4635    56Ø3            JMP I READHS
4636    1243    FIXUP,  TAD COMM1
4637    3645            DCA I INCOMM
464Ø    1243            TAD COMM1
4641    3646            DCA I INCOMM+1
4642    5232            JMP RETN
                        /
                        /
4643    34Ø1    COMM1,  READKB
4644    46Ø3    COMM1A, READHS
4645    2513    INCOMM, COMM
4646    3374            COMM34
4647    3455    OUTPUN, PUNOUT
465Ø    76ØØ    MIN2ØØ, 7600
4651    ØØØØ    TFM1,   Ø
4652    7766    CON1,   -12
4653    74Ø1            -377
4654    Ø173            173
```

NOTE: Once a symbol table has been read in via the high-speed reader, DDT is restored to its normal state. If the user types [R again, DDT will expect symbols from the teletype/low speed reader. If the user wishes to read in more than one symbol table tape with the high speed reader, he may NOP locations 4640 and 4641 of the patch, which restore the overlaid DDT location, and DDT will always expect additions to the symbol table to be input via the high speed reader. This patch cannot take full advantage of the high speed reader, for DDT still handles the input characters one at a time rather than buffering them as they are read.

An assembly listing of the patch follows:

PDP-8 DISK DDT (DEC-D8-CDE1 & DEC-D8-CDE2)

Failure to recognize overflow of the user symbol table

The Library version of Disk DDT does not correctly test for overflow of the user symbol table. The following patch will cause DDT-D, on occasion of symbol table overflow, to print a carriage return and line feed, and the new lower limit of the user symbol table, followed by a "↑" to indicate that it is waiting for a Control P (↑P).

Patch .DDT (DEC-D8-CDE2-PB) before saving it on the system device as follows:

LOCATION	FROM	TO
33Ø6	1361	5777
3353	Ø4Ø1	716Ø
3377	unused	4554
4554	"	4757
4555	"	176Ø
4556	"	5761
4557	"	3463
456Ø	"	3361
4561	"	33Ø7

PDP-8 DECTAPE COPY ROUTINE (DEC-08-YPTA-PB)

Problem in DECtape Copy Routine

DECtape Copy does not work if the field bits in the DECtape Status Register B are not Ø on entry to the program. The START key clears these bits, however, if the program is loaded by PS/8 from the disk file, these bits may be set.

The following patch will correct this problem:

LOCATION	FROM	TO
Ø242	745Ø	744Ø
Ø243	5247	1Ø44
Ø244	1Ø44	766Ø
Ø245	762Ø	52ØØ
Ø246	52ØØ	6774

PDP-8 DECtape FORMATTER (DEC-Ø8-EUFB-PB)

Patch to ignore DECtape flag when starting Pass 2.

The following patch corrects a problem in the current version of the DECtape FORMATTER. The patch is to ignore a spurious DECtape flag when the DECtape is doing a MOVE REVERSE, immediately after the mark and timing tracks have been written. Without the patch, the DECtape will halt, and it will be necessary to restart the program and rewrite the mark and timing tracks. Since this error only happens occasionally, the restart is usually not much of an inconvenience; but if the error should occur during batch formatting, say on the eighth tape, it would require doing pass 1 for all eight tapes again. Thus, users doing large-scale formatting should definitely implement the patch, for restart would be costly.

The patch is as follows:

LOCATION	OLD CONTENTS	NEW CONTENTS
1631	(2Ø21)	437Ø
177Ø	(unused)	ØØØØ
1771	(unused)	6771
1772	(unused)	741Ø
1773	(unused)	5222
1774	(unused)	2Ø21
1775	(unused)	577Ø
1776	(unused)	237Ø
1777	(unused)	577Ø

PDP-8 DECTAPE FORMATTER (DEC-Ø8-EUFB-PB)

Patch to Halt on receipt of unexpected interrupt

The current version of the PDP-8 DECtape Formatter does not correctly handle interrupts from unexpected devices (e.g., the LPØ8 Line Printer). The following patch will cause the Formatter, on receipt of an unexpected interrupt, to halt with

PC=Ø55Ø
MA=Ø547
MB=74Ø2

The user may then restart the program at Loc. 1ØØØ (the START key should clear the flag of the unexpected device).

The patch is as follows:

LOCATION	OLD CONTENTS	NEW CONTENTS
Ø545	(6Ø42)	5525
Ø547	(5364)	74Ø2
Ø633	(ØØØØ)	6Ø42
Ø634	(73ØØ)	6Ø31
Ø635	(1534)	5637
Ø636	(764Ø)	564Ø
Ø637	(5235)	Ø564
Ø64Ø	(6764)	Ø55Ø

PDP-8 DISK EDITOR (DEC-D8-ESAD-PB)

Two Problems in the DISK EDITOR have been found and corrected.

The first problem concerns rubbing out past the beginning of a line while in Search mode. Because it is caused by the contents of a previous text buffer, the problem occurs rarely. The correction to this problem is as follows:

LOCATION	OLD CONTENTS	NEW CONTENTS
1332	1014	5733
1333	3114	3342
3342	(unused)	1014
3343	(unused)	3114
3344	(unused)	3526
3345	(unused)	5746
3346	(unused)	1334
0053	3342	3347
0126	3342	3347

The second problem occurs when in Append mode. If an odd number of characters are put in a line terminated with a CR, and the user immediately starts rubbing out, the EDITOR will rub out even though no characters are on the current line. Again, this causes damage to the text buffer. If this occurs, please make the following correction:

LOCATION	OLD CONTENTS	NEW CONTENTS
1323	1173	1116
1362	5345	5344

PDP-8 DISK LINKING LOADER (DEC-08-A2C7)

Patch to allow the Linking Loader to recognize CONTROL C (↑C) when it is waiting for a FIELD specification.

The library version of the PDP-8 Disk Linking Loader fails to recognize a CONTROL C (↑C) when it is waiting for a FIELD specification following user selection of the L or O loading options; it treats ↑C as an illegal FIELD designation and responds with:

```
?
*OPT-
```

The following patch will correct this error and recognize ↑C as a call to the Monitor.

LOC	OLD CONTENTS	NEW CONTENTS
6065	(7200)	5374
6066	(5654)	0055
6174	(unused)	1266
6175	(")	7650
6176	(")	5451
6177	(")	5654

The patch may be toggled in after loading the Linking Loader into core. Once the patch is made the Loader may be resaved on the disk.

Patching programs before storing on the system device

The following steps apply to patching any disk/DECtape system program.

1. Set up the program (to be patched) for loading with the Disk System Loader. Answer all of its questions as usual with the exception of the starting address.

2. When the Loader requests the starting address (ST=), type 7636 for disk or 76Ø6 if the system device is DECtape. This will provide the halt referenced in step 4.

3. Load the program to be patched.

4. When loading is complete, the system will halt.

5. Now the desired patches may be made to the program in core.

6. Load address 7600, press START and control will return to the monitor.

7. Type the SAVE command as usual.

PDP-8 DISK LINKING LOADER (DEC-08-A2C7-PB)

Problem with deletion of temporary RIM format file

The current version of the Disk Linking Loader does not properly delete the temporary RIM format file it sometimes creates. This RIM format file is named .R.R.USER and is created when the user attempts to load data over the loader itself. This file is supposed to be deleted from the DN and SAM blocks upon user execution of an 'E' (EXIT with halt) or 'S' (START main program) option.

Due to improper handling of the DN and SAM block numbers, however, the temporary file is not correctly deleted from the Directory Name block, but remains there with a zero block-length specification. A further consequence of this problem may be the destruction of five words in one of the SAM blocks when the loader mistakenly zeros five words there, instead of the five word DN block entry for the RIM file.

The following patch will correct this problem and should be made to the Linking Loader before saving it on the system device:

LOCATION	OLD CONTENTS	NEW CONTENTS
425Ø	(4721)	5324
4324	(unused)	1Ø77
4325	(")	3332
4326	(")	4721
4327	(")	1332
433Ø	(")	3Ø77
4331	(")	5251
4332	(")	ØØØØ

Programming note on chaining in the PDP-8 Disk Monitor System

The following procedure illustrates the chaining of programs under the Disk Monitor System for the 8 Disk/DECtape system. First, save each overlay as a system program in the normal manner. Second, a call from one program to bring in the next program as an overlay must be done as follows:

1. In locations 74ØØ and 74Ø1 deposit the ASCII codes for the name of the system program to be loaded.
(Note: These ASCII codes are the eight bit code minus 24Ø)

2. In order to enter the routines which bring in the overlay, execute the following code:

```
                JMS I DYSKIO        /LINK TO MONITOR I/O
                3                   /READ
                1Ø                  /BLOCK #10 (DECTAPE #13)
                72ØØ                /CORE ADDRESS
                Ø
                HLT                 /ERROR RETURN
                JMP I  .+1          /LINK TO LOADER
                72Ø1
DYSKIO,         7642
```

The following example of a program that only brings in an overlay, called "TEST", may be helpful to many users.

```
                    *2ØØ
Ø2ØØ      7ØØØ      NOP
Ø2Ø1      7ØØØ      NOP
Ø2Ø2      4612      JMS I DYSKIO
Ø2Ø3      ØØØ3      3
Ø2Ø4      ØØ1Ø      1Ø
Ø2Ø5      72ØØ      72ØØ
Ø2Ø6      ØØØØ      Ø
Ø2Ø7      74Ø2      HLT
Ø21Ø      5611      JMP I .+1
Ø211      72Ø1      72Ø1
Ø212      7642DYSKIO,7642

                    *74ØØ
74ØØ      6445      6445      /TE WITH 24Ø SUB. FROM EACH CHAR
74Ø1      6364      6364      /ST
```

PDP-8 EAE FLOATING POINT PACKAGE (DIGITAL 8-25-F)

Patches to correct ALIGN Routine and check for largest negative mantissa

The current version of the PDP-8 EAE (Extended Arithmetic Element) Floating Point Package has an incorrect constant in the ALIGN routine for addition and subtraction. The constant only makes a difference when the two numbers being added vary greatly in magnitude. The following patch corrects the problem:

LOCATION	OLD CONTENTS	NEW CONTENTS
6167	7747	775Ø

The EAE Floating Point Package also fails to check for the largest negative mantissa (Octal 4ØØØ ØØØØ) after normalization. This number could cause erroneous results in subsequent calculations. The patch to correct this problem requires several changes to the EAE package itself, plus a fifteen word patch which can be inserted anywhere in the same memory field as the EAE Floating Point Package. In the listing that follows, the patch has been placed just before the start of the Interpreter (i.e. just before loc. 56ØØ). This patch would have to be moved for use with other versions of the EAE Packages (i.e. for packages 2, 3 or 4). The necessary changes to the EAE package are:

LOCATION	OLD CONTENTS	NEW CONTENTS	
66Ø7	326Ø	3367	
6655	226Ø	2367	
6657	56ØØ	5766	
666Ø	ØØØØ	56ØØ	
6766	unused	5561	(address of first word of patch)

The following is the fifteen location patch which may be placed anywhere in the appropriate memory field. Note, location 6766 must contain the address of the first word of this patch.

LOCATION	CONTENTS
5561	1Ø46
5562	764Ø
5563	5777
5564	1Ø45
5565	751Ø
5566	7Ø41
5567	77ØØ
557Ø	5777
5571	2Ø44
5572	7ØØØ
5573	1Ø45
5574	713Ø
5575	3Ø45
5576	5777
5577	666Ø

PDP-12

Problem with EAE

A software (hardware) problem has been reported with EAE (KE12). The instructions CLA (7601) and NMI (7411) are defined as being micro programmable, but due to a hardware problem they do not function properly when mico programmed. Hence, the instructions CLA and NMI should be used as two separate instructions.

No correction is planned for the hardware.

PDP-8 SYMBOLIC EDITOR (DEC-08-ESAC)

Problem with missing line feed

When using the Next (N) command for output to the high speed punch, a timing problem causes the final line feed (ASCII code 212) to be lost, ending the tape with a carriage return (ASCII code 215). This difficulty occurs in the current version of the Editor (DEC-08-ESAC). The problem may be corrected with the following patch:

Change: COM1,JMP FORM to COM1,JMP FORM+1

LOCATION	FROM	TO
13ØØ	5216	5217

PDP-8 SYMBOLIC EDITOR (DEC-08-ESAC)

Problem with rubbing out past beginning of lines

There is a problem with the current library version of the Symbolic Editor such that under certain conditions, the user can delete characters from right to left past the beginning of a line. This usually results in the Editor losing track of the proper line sequences. Making the following patch before starting the Editor will eliminate the problem:

LOCATION	OLD CONTENTS	NEW CONTENTS
ØØ57	1624	1631
Ø115	1624	1631
1324	1Ø13	5725
1325	3124	1624
1624	unused	3515
1625	"	1Ø13
1626	"	3124
1627	"	563Ø
163Ø	"	1326

EDUSYSTEM 20 AND 50

Problem with 8/e KL8-e (Teletype Interface)

When running on the PDP-8e, both Edusystem 20 and 50 (TS8-E) sometimes have keyboards "go dead" for apparently no known reason.

The problem is apparently caused by the "build-up" of static electricity in terminal users' metal chairs, and the subsequent discharge on contact with the teletype. The problem only exists with respect to terminals connected directly to the Processor, and one recommendation for overcoming the problem is to use wooden chairs.

The problem may be compensated for by means of software. A patch for TS8-E Version 8.22B follows. A correction for Edu. 20 is currently not available, however a new version will be available in the near future which will include a software fix.

TS8/E VERSION 8.22B

```
            /PATCH TO TS8/E  VERSION 8.22B
            /RE-ENABLES KEYBOARD INTERRUPTS
            /TO CORRECT FOR KL8-E HARDWARE BUG

  Ø114      KDEV=Ø114          /POINTS TO BEGINNING ON KEYBOARD DEVICE CODES
  Ø113      TDEV=Ø113          /POINTS TO END OF KEYBOARD DEVICE CODES

  2412                *2412

2412  4773  L2TIME, JMS  I PATCHL    /TO PATCH - WAS "ION"

  2573                *2573

2573  7152  PATCHL, PATCH
              NULJOB,                /NORMAL NULL JOB


  715Ø                *715Ø

715Ø  ØØØØ  PATCH,  Ø
7151  2374            ISZ PATCNT     /ONLY EVERY SO MANY SYSTEM TICKS
7152  5372            JMP PATCH2     /OTHERWISE, JUST RETURN TO NORMAL

7153  1375            TAD PATINI     /RESET COUNTER
7154  3374            DCA PATCNT
7155  1114            TAD KDEV       /SET POINTER TO KEYBOARD DEVICE CODES
7156  3376            DCA PATPTR
 .57  2376  PATCH1,   ISZ PATPTR     /ON TO NEXT KEYBOARD
```

```
7160    1776            TAD I PATPTR        /GET DEVICE CODE
7161    1377            TAD C6005           /MAKE IT INTO "KIE"
7162    3364            DCA .+2             /SET TO EXECUTE IT
7163    7001            IAC                 /"1" IN AC TO ENABLE
7164    0000            .-.
7165    1376            TAD PATPTR          /CHECK FOR END OF DEVICE CODES
7166    7041            CIA
7167    1113            TAD TDEV
7170    7700            SMA CLA
7171    5357            JMP PATCH1          /NOT YET - KEEP GOING

7172    6001    PATCH2, ION                 /TURN INTERRUPT BACK ON
7173    5750            JMP 1 PATCH         /AND RETURN TO NORMAL


7174    7740    PATCNT, -40
7175    7740    PATINI, -40
7176    0000    PATPTR, 0
7177    6005    C6005,  6005

                $$$$$$
```

PDP-8 FLOATING POINT PACKAGE(DEC-08-YQ2B,DEC-08-YQ4B)

Problem in Output Controller when Outputting Zero

There is a problem in the Output Controller of the PDP-8 FLOATING POINT PACKAGE which will cause an incorrect number of digits to be printed when the number being output is zero. The Controller does not always print the correct number of zeros, but the printed result will never be other than zero. Since the Controller is present in FLOATING POINT PACKAGES 2 (two) and 4 (four), the patch which follows should only be implemented by users working with these version -- DEC-08-YQ2B-PB and DEC-08-YQ4B-PB.

The Patch is:

LOCATION	OLD CONTENTS	NEW CONTENTS
5425	(2214)	5626
5426	(2214)	7365
5445	(22ØØ)	7ØØØ
7365	(unused)	1775
7366	(unused)	765Ø
7367	(unused)	2Ø44
737Ø	(unused)	2774
7371	(unused)	2774
7372	(unused)	5773
7373	(unused)	5427
7374	(unused)	5414
7375	(unused)	5572

Note: This patch is the same for both packages 2 and 4.

PDP-8 FLOATING POINT PACKAGES 2 and 4 (DEC-08-YQ2B-PB,DEC-08-YQ4B-PB)

Patch to Make Output Controller Recognize Loc. 55 as a Switch Governing the Printing of a Carriage Return/ Line Feed.

The Output Controller in the PDP-8 FLOATING POINT PACKAGE does not correctly recognize location 55 as a switch governing the printing of a CR/LF after output. The switch is correctly recognized in packages 1 and 3. To make packages 2 and 4 recognize the switch, make the following patch: (same for both versions)

LOCATION	OLD CONTENTS	NEW CONTENTS
72Ø7	(72ØØ)	1Ø55

FPP FORTRAN IV RUN-TIME SYSTEM-FORRTS.FT (V4)

Patch to correct erroneous results from unfilled format fields

In a formatted READ, if rubout is used to delete characters and they are not replaced to fill the format field, an erroneous value results. For example, for a format field F5.2, typing 1.234\\\ produces 1.03. The following fix solves the problem.

Use the program FPIP as follows:

```
.R FPIP                          Paper tape and editing program
*DSK:FORRTS.FT<1$                Systems device;$=altmode
R,11                             Read block 11
O,315                            Open relative location 315
7001/5341                        Change to 5341
O,341                            Open relative loc. 341
0000/1774                        Change 341 to 1774
0000/3350                        Change 342 to 3350
0000/1114                        Change 343 to 1114
0000/3750                        Change 344 to 3750
0000/7001                        Change 345 to 7001
0000/5316                        Change 346 to 5316
W                                Write the block
E                                Exit FPIP
↑C                               Return to Command Decoder
```

The underlined portions are printed by FPIP and are the current contents of the locations. Note that FPIP is a new OS/8 program whose 2 main functions are to read in paper-tape patches and to permit typing in short fixes (similar to ODT) for the FPP FORTRAN IV files.

FPP FORTRAN IV-LOAD.SV (V13)

Patch to prevent the omission of overlay information

An input line to the loader (LOAD.SV) of just * /O (no input files) causes all preceding overlay information to be disregarded.

The following patch eliminates the problem.

Use OS/8 ODT to add the changes.

LOCATION	OLD CONTENTS	NEW CONTENTS
13152	3421	745Ø
13153	2Ø21	5741
13154	3Ø22	3421
13155	5741	2Ø21
13156	Ø	3Ø22
13157	Ø	5741

FOCAL 1969 (DEC-08-AJAE)

Failure to read user programs on a PDP-8/S

FOCAL 1969 sometimes fails to read user generated tape programs via the High Speed Reader on an 8/S. This problem is due to a timing problem in the reader delay loop. The following patch will correct the problem:

Change location 6322 after executing the dialogue:

OLD CONTENTS	NEW CONTENTS
7Ø4Ø (CMA)	112Ø (TAD M5)

Note: The dialogue attempts to reduce the time-out delay by changing TAD M2Ø to CMA, but this is not correct.

PDP-8 FOCAL LIBRA System (Multi-user overlay)

Patch to accept more common variables

FOCAL LIBRA can be modified to accept more common variables by using the following patch after the initial dialogue:

LOCATION	FROM	TO
0035	4560	4553
6362	4565	4560
6363	7773	7771
6364	0005	0007

FOCAL 1969 (DEC-08-AJAE)

Patch to correct Input-Buffer-Overflow

To avoid input-buffer-overflow while reading in a tape, remove the echo using the following patch:

LOCATION	OLD CONTENTS	NEW CONTENTS
2163	4551	7ØØØ

To restore the echo, insert the old contents of location 2163.

FOCAL 1969 (DEC-08-AJAE)

Patch to correct one of the SINE Constants

To correct one of the SINE Constants, use the following patch:

LOCATION	FROM	TO
5333	24Ø1	25Ø1

PDP-8 FOCAL (DEC-08-AJAE-PB) with 8K Overlay

Patch to 8K FOCAL with DISK/DECtape Monitor

After loading the 8K Overlay use the following patch if you are having problems saving, restoring or running 8K FOCAL Programs with the DISK/DECtape Monitor System.

LOCATION	FROM	TO
7525	6213	6212

FOCAL 1969 (DEC-08-AJAE)

Note to users of FOCAL

To read data tapes on the TTY reader, use the following patch to remove the interrupts:

LOCATION	FROM	TO
63	2676	1354
64	2666	2414
2732	6ØØ1	5336
2762	6Ø46	7ØØØ

To allow use of the L command while the interrupt is off, the following patch should also be made:

LOCATION	FROM	TO
7522	1016	5325

Note: These patches disable high-speed readers unless the following patch is also made:

LOCATION	FROM	TO
6324	1Ø37	6Ø11
6325	77ØØ	741Ø

PDP-8 FOCAL '69 (DEC-08-AJAE)

Patch to use LINC-8 display (ON THE LINC-8 ONLY)

To use the LINC-8 display with FOCAL '69 use the following code which should be assembled with PAL III. This binary tape must be loaded after loading FOCAL. This patch is not intended for use on the PDP-12.

```
                    /DISPLAY PATCH FOR FOCAL  69
                    START=4500
                    /VALUE OF START IS DETERMINED BY TYPING   L   TO FOCAL
                     AFTER
                    /INITIAL DIALOGUE.  THE LAST VALUE TYPED MINUS 31 OR
                     GREATER
                    /THAT DOES NOT CROSS PAGE BOUNDARY
                    XDYS=1142
                    *XDYS +1
1143      3757      DCA I X
                    *XDYS+11
1153      6167      6167
1154      7200      CLA
1155      5756      JMP I PATCH
1156      4551      PATCH,LINDIS
1157      4550      X,SAVX
                    *BOTTOM
0035      4547      START -1
                    *START
4550      0000      SAVX,0
4551      1371      LINDIS,TAD LINCPC /LINC DISPLAY ROUTINE
4552      6165      6165
4553      7200      CLA
4554      1775      TAD I SAVFOC
4555      3374      DCA SAVLOC
4556      1350      TAD SAVX
4557      3775      DCA I SAVFOC
4560      1376      TAD ENABL
4561      6141      6141
                    EFUN3I=136
4562      1377      TAD TWLV
4563      6141      6141
4564      7000      NOP
4565      7200      CLA
4566      1374      TAD SAVLOC
4567      3775      DCA I SAVFOC
4570      5536      JMP I ENFUN3I /RETURN TO FOCAL
4571      4572      LINCPC,NEXT
4572      0141      NEXT,141      /LINC CODE DISPLAY 1
4573      0000      0             /LINC HALT
4574      0000      SAVLOC,0
4575      4001      SAVFOC,4001
4576      0010      ENABL,10
4566      0002      TWLV,2
                    BOTTOM=35
```

```
BOTTOM   0035
EFUN3I   0136
ENABL    4576
LINCPCP  4571
LINDIS   4551
NEXT     4572
PATCH    1156
SAVFOC   4575
SAVLOC   4574
SAVX     4550
START    4550
TWLV     4577
X        1157
XDYS     1142
```

PDP-8 FOCAL '69 Machine Language Subroutine

FNEW functions must be written in assembly language in patch form and loaded over the FOCAL program. These may be loaded after FOCAL has successfully loaded, and do not affect the "FOCAL" program the user has previously entered.

Subprogram to operate on an indefinite number of arguments.

Called from FOCAL by: SET = FNEW (A,B,C,...,N); TYPE Z where A,B,etc. have been previously defined and are to be operated upon to yield Z. Subprogram must be assembled as a patch and loaded over FOCAL.

```
START = (SEE NOTE)
*BOTTOM
 FNEW-1
*FNTAB+14
 FNEW
*START
 FNEW,TAD LIST-1
      DCA 16              /PUT ADDRESS OF LIST INTO AUTOINDEX REG.
      DCA KOUNT
      ISZ KOUNT
      JMS I INTEGER       /FETCH NEXT ARGUMENT
      DCA I 16            /STORE IN LIST
      PUSHJ
      ARG                 /GET NEXT CHARACTER
      SKP                 /RETURN IF NOT A COMMA
      JMP FNEW+3          /RETURN IF COMMA WITH NEXT ARG. IN FLAC
      CLA CLL             USER CODING TO OPERATE ON N ITEMS
      .                   STORED IN LIST.  KOUNT CONTAINS THE NUMBER
      .                   OF ITEMS,N.  THE RESULT MUST BE STORED
      .                   IN INTEGER FORM IN ANS.
      CLA CLL
      TAD ANS             /CODING TO SET ANS INTO FLAC
      CLL RAR
      DCA FLAC+1
      RAR
      DCA FLAC+2
      TAD C14
      DCA FLAC
      JMP I EFUN3I        /CHECK FOR RIGHT PAREN,NORMALIZE FLAC,RETURN
C14,14
KOUNT,Ø
ANS,Ø
      LIST-1
LIST, Ø
      Ø
      .
      .                   /AS LONG AS ARGUMENT LIST WILL BE
      .
```

```
START = (SEE NOTE)
BOTTOM
FNEW-1
*FNTAB+14
FNEW
*START
FNEW,DCA SUM              /CLEAR SUM
       JMS I INTEGER      /BRING FIRST (NEXT) CHAR. INTO ACCUMULATOR
       TAD SUM
       DCA SUM            /UPDATE SUM
       PUSHJ
       ARG                /GET NEXT CHAR.
       SKP                /RETURN IF NOT A COMMA
       JMP FNEW+1         /RETURN IF COMMA WITH NEXT ARG IN FLAC
       CLA CLL
       TAD SUM            /FINAL SUM SET INTO FLAC FOR
       CLL RAR            /RETURN FROM SUBPROGRAM
       DCA FLAC+1
       RAR
       DCA FLAC+2
       TAD C14
       DCA FLAC
       JMP I EFUN3I       /CHECK FOR RIGHT PAREN,NORMALIZE FLAC,RET.
C14,14                    /FLAC,RETURN
SUM,   Ø
ARG,   TAD CHAR           /CODING TO FETCH NEXT CHAR.
       TAD MCOMMA         /RETURN TO CALL+3 (OR CALL+2)
       SZA CLA            /IF CHAR IS (IS NOT) A COMMA
       JMP  .+4 .
       PUSHJ
       EVAL-1
       IAC
       POPJ
MCOMMA,7524

INTEGER=53                /THESE LOCATIONS MAY VARY
PUSHJ-4540                /WITH THE VERSION OF FOCAL
EVAL=1613                 /FOR FOCAL '69 DEC-08-AJAE-PB
FLAC=44                   /THESE DEFINITIONS APPLY
CHAR=66
EFUN3I=136
BOTTOM=35
FNTAB=374
```

NOTE: CODING MAY OCCUPY CORE FROM 3220-4577. THIS AREA IS OCCUPIED BY STORAGE LISTS AND PUSH-DOWN LISTS, SO CODING SHOULD OCCUPY THE TOP OF THIS REGION IN CORE SO AS NOT TO SEVERELY LIMIT THE WORKING STORAGE AREA. IF FEXP, FLOG, AND FATN ARE DELETED, THE USER HAS AVAILABLE LOCATIONS 3220-5177. IF IN ADDITION TO ABOVE, FSIN AND FCOS ARE DELETED, THE USER HAS AVAILABLE 3220-5232. TO DETERMINE THE VALUE FOR "START" USE THE "L" COMMAND IN FOCAL. START=LAST VALUE PRINTED MINUS LENGTH OF PATCH STARTING AT *START.

```
ARG,    TAD CHAR          /CODING TO FETCH NEXT CHAR.
        TAD MCOMMA        /RETURN TO CALL+3 (CALL+2)
        SZA CLA           /IF CHAR. IS (IS NOT) A COMMA
        JMP  .+4
        PUSHJ
        EVAL-1
        IAC
        POPJ
MCOMMA,7524

INTEGER=53                /THESE VALUES MAY VARY
PUSHJ=4540                /WITH THE VERSION OF FOCAL USED
EVAL=1613                 /FOR FOCAL '69 DEC-08-AJAE-PB
FLAC=44                   /THESE DEFINITIONS APPLY
CHAR=66
EFUN3I=136
BOTTOM=35
FNTAB=374
```

NOTE: CODING MAY OCCUPY UPPER END OF STORAGE USE FOR TEXT AND PUSH-DOWN LISTS. THIS AREA OCCUPIES CORE LOCATIONS:

3220-4577	with all function
3220-5177	with FEXP,FLOG,FATN,
3220-5232	FSIN,FCOS and above deleted.

TO DETERMINE THE VALUE FOR "START" USE THE "L" COMMAND IN FOCAL. "START" = LAST VALUE PRINTED MINUS LENGTH OF PATCH STARTING AT *START.

On the following page is a subprogram which sums an indefinite number of arguments. It is called from FOCAL by:
SET Z = FNEW (A,B,C,.....,N) :TYPE Z where A,B, etc. have been previously defined. This subprogram must be assembled as a patch loaded over FOCAL.

FOCAL 1969 (DEC-08-AJAE)

Programming Note

"TYPE #" issues a carriage return plus a FORM feed to allow the carriage return to get back to the left margin. On certain models of teletypes, multiple line feeds are generated upon reception of a form feed. The following patch to replace form feed (214) with a null character (216) should be used if you have this type of TTY:

LOCATION	OLD CONTENTS	NEW CONTENTS
125Ø	7ØØ1	7Ø4Ø

FOCAL 1969 (DEC-08-AJAE)

Patch to correct rounding error

A rounding error has been discovered in FOCAL where a remainder of 4 in the first digit discarded causes the last digit printed to be rounded upwards. To correct this problem, the following patch should be used:

LOCATION	FROM	TO
5525	ØØØ5	ØØØ4

FOCAL 1969 (DEC-08-AJAE)

Patch to allow automatic restart with L command

The following patch will automatically restart FOCAL when using the L command (upon completion of the four locations printout).

LOCATION	FROM	TO
7526	55Ø4	5177

Note: This patch must be used when the Disk Monitor System is not in use.

FOCAL 1969 (DEC-08-AJAE) with CLINE & PLOTR

Patch to correct circle

If the radius of a circle being plotted is an exact power of 2, it is mistakenly interrupted as Ø and a string of dots is drawn in place of the circle.

The following patch should be used (according to the overlay in use) :

For CLINE:

LOCATION	FROM	TO
5Ø6Ø	1Ø46	1Ø45

For PLOTR:

LOCATION	FROM	TO
5Ø32	1Ø46	1Ø45

FOCAL 1969 (DEC-08-AJAE)

Patch for use on LAB-8

A problem has been discovered when using FOCAL on a LAB-8 system. A possible enabling of the clock interrupt may occur. To prevent this, the following patch should be made before the initial dialogue:

LOCATION	FROM	TO
4377	6Ø12	7ØØØ

FOCAL 69 (DEC-08-AJAE and DEC-08-AJ6E)

Patch to FOCAL-69 LIBRA on an RFØ8 disk.

Location Ø15Ø on field Ø is being overlaid by DISKIN accidentally. To correct this problem:

1. Load FOCAL into field 1
2. Load LIBRA into field Ø
3. Change location 2611 in field Ø from 5612 to 74Ø2
4. Start at 2ØØ
5. Answer all the questions and when the computer halts change location Ø15Ø in field Ø from a 3Ø53 to a 6641
6. Start at 2ØØ

PDP-8 FOCAL'69 and QUAD/LIBRA

Saving FOCAL'69 and QUAD/LIBRA on a DIAL LINCtape

The following is the procedure for loading FOCAL'69 and its overlays (QUAD/LIBRA) on a DIAL LINCtape as one binary so it may be loaded into core directly via the LO command.

```
 Start LAP6-DIAL-MS          (LS=7Ø1; RS=731Ø)
→ZE                          (zero binary working area)
→PI                          (load PIP)
  B                          (binary mode)
  T                          (input - TTY)
  LØ; FOCAL 4K               (output - LINCtape Ø)
  D                          (CNTL D - return to LAP6-DIAL
                              after reading in both parts of FOCAL)
→ZE                          (zero binary working area)
→PI                          (load PIP again)
  B                          (binary mode)
  T                          (input - TTY)
  LØ; QUADPTØ8               (output - LINCtape Ø)
  D                          (CNTL D - return to LAP6-DIAL after
                              QUAD overlay is saved)
→ZE                          (zero binary working area)
→AB ØØØØ,1,FOCAL 4K,Ø        (this will add back the binary of
                              FOCAL and sets the bit map so it
                              will be loaded into FIELD 1)
→AB QUADPTØ8,Ø               (this adds the binary for QUAD to
                              the FOCAL binary and sets up the
                              whole bit map)
→SB ANYTHING,Ø,P             (This will save the two binaries
                              as one file and when loaded will
                              start in 8 mode, location 2ØØ)
```

Notes:

LAP6-DIAL-MS will not load the last page in Field 1 (loc. 176ØØ - 17777) also, LIBRA stores some data into those locations. In order to load a LAP6-DIAL file (FOCAL plus LIBRA) which uses this last page, you must use the DIAL-V2 Loader supplied on LINCtape DEC-12-SE2E-UO, not the DIAL-MS Loader. This does not apply to QUAD since it does not use that last page of core.

Please see article FOCAL-14 for a patch to LIBRA RFØ8 version.

When you PIP a binary program onto a LINCtape/Disk, the binary working area must be zeroed first, since PIP stores the file there before it transferrs it into its stored area. Therefore, when you later do an ADD Binary everything will work properly. (This is described in the LAP6-DIAL manual, DEC-12-SE2D-D, pages 4-5 to 4-7).

FOCAL'69

Problem when running FOCAL'69 with API on PDP-12

FOCAL'69 (DEC-08-AJAE), while executing its initialization code, issues a 6762 (clear TCØ1 status register A) which is also used in the new PDP-12 API (Automatic Priority Interrupt) hardware. Therefore, if you are running on a PDP-12 with API this instruction must be NOP'ed.

To correct this problem, do the following:

1) Set left switches = Ø2ØØ
2) Set F stop key
3) →LO FOCAL4K,Ø (from DEC-12-SE2E-UO)
4) When the machine halts, set left switches = 4376; right switches = 7ØØØ; press the fill key
5) Make sure the machine is in PDP-8 mode, set the left switches = Ø2ØØ
6) Press start left switches switch

To correct the DIAL System Tape (DEC-12-SE2E-UO) change the following tape block:

BLOCK	WORD	FROM	TO
237	376	6762	7ØØØ

```
                /PATCH TO FOCAL 1969 (AJAE)
                /TO CURE BUFFER OVERFLOW PROBLEM (PDP-8/E ONLY)
                /MAKES USE OF EXTRA 8/E KEYBOARD IOT (KCF)
                /SO MAKING IT IMPOSSIBLE FOR OVERFLOW TO OCCUR
                /PATCHES FOCAL INTERRUPT PROCESSOR
                /LISTING PAGE 44
                /AND FOCAL TTY INPUT ROUTINE
                /LISTING PAGE 45
                 OPATCH=2414
                 INBUF=34
                 P177= 106
                 MBREAK= 2602
                 C200= 123
                 EXIT= 2646
                 RECOVR= 2740
                 SIN=2662
                 INRET=2675
                 KCF=6030
                 *2627
                /PATCH TO INTERRUPT PROCESSOR
2627    6034             KRS              /INPUT CHARACTER
2630    6030             KCF              /CLEAR FLAG
2631    0106             AND P177         /IGNORE BIT 8
2632    7450             SNA              /BLANK?
2633    5245             JMP EXIT-1       /YES-GO INITIATE NEXT READ
2634    1123             TAD C200         /SET BIT 8 ON
2635    3262             DCA SIN
2636    1262             TAD SIN
2637    1202             TAD MBREAK
2640    7650             SNA CLA          /CTRL C?
2641    5340             JMP RECOVR       /YES
2642    1262             TAD SIN
2643    3034             DCA INBUF        /STORE CHARACTER
2644    7410             SKP
2645    6032             KCC              /INITIATE NEXT READ- CHAR. WAS A BLANK
                *2673
2673    5674             JMP I .+1        /PATCH TO INPUT ROUTINE
2674    2414             OPATCH
                *OPATCH
2414    3034    OPATCH   DCA INBUF        /CLEAR INPUT BUFFER
2415    6032             KCC              /INITIATE NEXT READ
2416    1621             TAD I XOUTL      /GET CHARACTER
2417    5620             JMP I .+1        /RETURN TO INPUT ROUTINE
2420    2675             INRET
2421    2676    XOUTL,   2676
                $
```

FOCAL 1969 (DEC-Ø8-AJAE)

Optimum use of the PDP-8/E

The following patch uses the KCF instruction of the PDP-8/E to eliminate input buffer overflow. The patch should be assembled with PAL III so that the binary can be loaded easily each time FOCAL is loaded.

PDP-12 FOCAL-12

Patch to allow FOCAL-12 to save files on a 1600_8 block LINCtape

To allow FOCAL-12 (on LINCtape DEC-12-SE2E-UO) to save files on a 1600_8 block LINCtape, make the following changes to the binary:

FOCAL-12 (DEC-12-AJAA)

23_8 rd block of the actual binary (starting block in index + 23_8)

WORD	FROM	TO
353	6777	6177

24_8 th block of the actual binary (starting block in index + 24_8)

WORD	FROM	TO
161	7ØØØ	6ØØØ

NOTE: Please notice the above change from header block to starting block number, when you display the index of a LAP6-DIAL tape.

PDP-12 FOCAL-12

Note on FADC instruction

It is important to remember that FOCAL-12's FADC instruction references the AD-12 channels as if they were numbered decimally.

Therefore, channel 10 = FADC (8), channel 11 = FADC (9), etc.

PDP-8 4K FORTRAN Compiler (DEC-08-AFC1)

Procedure to restore locations destroyed by SYMBOLPRINT (DEC-08-AFA2)

The PDP-8 4K FORTRAN compiler is designed to be restarted for successive compilations and not require reloading after compiling each source program. The PDP-8 FORTRAN Symbolprint program, which is read in over the compiler to use the information left in its tables and counters, resides in the compiler's input statement so that this restart capability will not be lost.

However, the use of Symbolprint imposes two restrictions on the restart facility:

1. The compiler must be reloaded after each use of Symbolprint if the user chooses the switch option to indicate that his FORTRAN source program contains DECtape I/O statements. This is a design restruction due to lack of free space in the compiler. A portion of the additional code to compile DECtape I/O statements is destroyed by Symbolprint.

2. The contents of three locations in the compiler are changed by Symbolprint during execution. They must be restored or subsequent compilations will yield incorrect results, possibly without producing any error messages.

   The locations and their proper contents are:

LOCATION	CONTENTS
1Ø	ØØØØ
11	Ø6Ø3
12	Ø6Ø4

   These values must be restored by the following procedure after running Symbolprint and before starting the next compilation.

Set	switch	register	to	0010	depress	LOAD ADDRESS
"	"	"	"	0000	"	DEPOSIT
"	"	"	"	0603	"	DEPOSIT
"	"	"	"	0604	"	DEPOSIT

   Restart the compiler as usual.

PDP-8 4K Fortran Compiler and Operating System
(DEC-08-AFC1-PB and DEC-08-AFC3-PB)

Patch to Compiler Input Overflow and Input Conversion Limitations

The library version of 4K Fortran Compiler (DEC-08-AFC1-PB) uses a mask of 74ØØ to check for overflow in the high order mantissa word during floating point input conversion. The correct mask is 76ØØ. To patch the compiler, load it using the binary loader, then DEPOSIT 76ØØ into location 5515 (old contents of 5515 was 74ØØ). The compiler may now be started as usual.

Users should not attempt to input floating point constants of more than six decimal digits, either in the Fortran source program or via the run-time ACCEPT statement, as in general such constants will not be handled correctly. The results of including a constant of the form XXXXXXX. in the source program are unpredictable, while a constant of the form .XXXXXXXE±XX will be represented correctly to six decimal places. Attempting to input a constant of the form XXXXXXX. via the run-time ACCEPT statement, may cause an error in the calculation of the Exponent, while inputting the same constant as XXXXXX.XE+1 via the ACCEPT statement, will cause the exponent to be calculated correctly, and the mantissa to be correct to 6 decimal places. Users should be especially aware of the fact that different representations of the same constant (7 or more digits) may cause different results when these representations are input to the compiler, or when input via the run-time ACCEPT statement.

4K FORTRAN COMPILER (DEC-08-AFC1-PB)

Problem with mode of variables.

The current version of the 4K FORTRAN COMPILER (DEC-08-AFC1-PB) is inconsistent in its handling of variables when the mode (integer of floating point) associated with the variable name differs from the mode specification in a FORMAT statement. Several examples will illustrate this inconsistency:

A)
```
1;    FORMAT (E)
      ACCEPT 1,I
      TYPE   1,I
```

In the above example, if the number 1.23 followed by a terminator is typed in response to the ACCEPT statement, the decimal part will be lost and the number will be stored as an integer on input (only one word of storage will be allocated the variable I - the number will be input as a floating point number, but then it will be converted to fixed point and stored as an integer). On output, the print-out will be:

```
+0.100000E+1
```

so the integer mode associated with the variable name I has taken precedence over the format specification on input but not on output.

B) In the case

```
1;    FORMAT (I)
      ACCEPT 1,A
      TYPE   1,A
```

the integer mode specification in the FORMAT statement takes precedence on input and output.

C) Inputting the number 1.33 followed by a terminator to the following:

```
1;    FORMAT (I,E)
      ACCEPT 1,A,B
      TYPE   1,A,B
```

will yield as output: +1 +0.330000E+2

for the decimal point is recognized only as a terminator of the input for the integer variable A (FORMAT statement has precedence as in example B, above), and 33 is taken as input for the floating point variable B. Had the program looked as follows:

```
1;    FORMAT (E,I)
      ACCEPT 1,I,J
      TYPE   1,I,J
```

then inputting 1.33 followed by a terminator would not satisfy the ACCEPT statement, for 1.33 would be read in as a floating point number, then converted to fixed point and stored as integer +1 (as in example A), the operating system would then still be waiting for more input; if the number 2 is now typed, followed by a terminator, the ACCEPT statement would be satisfied and output would be as follows:

```
+0.100000E+1    +2
```

(again output for the variable I reacts as described in example A).

PDP-8 4K FORTRAN COMPILER AND SYMBOLPRINT
(DEC-08-AFC1-PB and DEC-08-AFA2-PB)

Problem with SYMBOLPRINT typeout of highest address used for storage of interpretive code

4K FORTRAN SYMBOLPRINT types out a list of symbols and corresponding values and also types out the upper limit of core-storage for the compiler-generated interpretive code, and the lower limit of core used for data storage. The printed value of the first of these numbers, the upper limit of interpretive code, is higher than the actual value of the limit. This is due to a problem in the FORTRAN COMPILER. To have the correct value printed by SYMBOLPRINT, make the following patch to the COMPILER before compiling the FORTRAN source program:

LOCATION	FROM	TO
433Ø	(42ØØ)	4251

PDP-8 FORTRAN-D (DEC-D8-AFA1-6)

Suggested method for saving FORTRAN-D object programs.

The following method for saving FORTRAN-D object programs enables the user to call his program with the FORTRAN-D operating system as a Monitor system program.

1. Compile the FORTRAN-D source program as usual.

2. Load and run the compiler output under FOSL.

3. At the end of a successful run, control will return to the Monitor. The user should then call FOSL to reload the compiler output as if to run the program again.

4. When FOSL has loaded the program (and subroutines, if any) it will type

   READY
   ↑

   The user should type CONTROL C (↑C) to return to the Monitor.

5. A core image of the user's program in FOSL may not be saved on disk as a system program by typing

   SAVE NAME!Ø-7577;5Ø43

   Note that the starting address must be 5Ø43.

6. When the Monitor has returned control to the user, the core image may be called at any time as a system program. On starting, the program will type

   READY
   ↑

7. CONTROL P (↑P) will initiate execution. When execution is complete the program will type "!" and return to the Monitor.

Users who run FORTRAN-D with binary subroutines will find this method especially time saving.

PDP-8 FORTRAN-D (DEC-D8-AFA2)

Patch to correct problem in floating point input routine.

The floating point input conversion routine in .FT. fails to correctly compile decimal numbers which have more than six significant digits and are above a certain magnitude. This problem results from an error in masking to check whether accepting another decimal digit will cause an overflow. The mask is now set at 7400 (octal) and, therefore, applies to the top four bits of the double precision word. However, the top five bits should be interrogated as the leading sign bit must not be changed. To correct this situation, the mask at location 4715 (octal) should be set to 7600 (octal). A similar conversion routine in the operating system does use a mask of 7600. This patch will be implemented in the next revision of FORTRAN-D. Until such time see the article DISK MONITOR SYSTEM 4 for a description of how to make the patch to location 4715 of the Compiler (.FT.).

PDP-8 FORTRAN-D (DEC-D8-AFA3 and AFA4-PB)

Patch to allow non-system DECtape I/O.

In order to perform I/O on a DECtape other than that being used as the system device, it is necessary to make the patch given below to FOSL and .OS. of the Fortran-D Operating System.

Without this patch, I/O can only be handled via the teletype, the high speed reader/punch, or the system device. With the patch, I/O is enabled on DECtape unit 7 rather than on the DECtape specified as the system device.

The procedure for loading the Operating System, patching it, and saving it is outlined as follows.

1. Load the operating system loader (FOSL) using Loader in one pass with a 7636 argument to ST=.

```
.LOAD ↵
*IN-R: ↵
*
*OPT-1
*ST=7636 ↵
↑↑
```

After FOSL has been completely loaded, the Loader will halt at location 7636 enabling you to enter the following patch through the switch register.

Instructions to allow reading from DECtape 7.

LOCATION	NEW CONTENTS	
166	27ØØ	
27ØØ	1377	
27Ø1	3766	
1224	7Ø3	/DECtape unit 7 input (603 for input from unit 6)

Instructions to allow writing onto DECtape 7.

27Ø2	1375	
27Ø3	3774	
27Ø4	5773	
1324	7Ø5	/DECtape unit 7 output (605 for output to unit 6)

Storage cells for the above changes:

LOCATIONS	NEW CONTENTS
2773	263
2774	5353
2775	705
2776	5331
2777	7Ø3

Once the patch has been made, load address 76ØØ and press START to return to the monitor. FOSL may then be saved in the following manner.

.SAVE FOSL! Ø-1577,27ØØ;2ØØ↵

2. The operating system interpretive and arithmetic package (.OS.) should now be loaded by using the Loader in one pass and specifying 7636 as the argument to ST=.

.LOAD ↵
*IN-R:↵
*
*OPT-1
*ST=7636↵
↑↑

The loader will again halt at location 7636 and the following patch must be made to allow writing on DECtape 7.

LOCATIONS	NEW CONTENTS
5155	1377
5177	7Ø3

When the patch has been made, load address 76ØØ and press START. Control will return to the monitor and you may now save .OS. as .SAVE .OS.! Ø-5177; The following page shows an example program and its results to prove this new method of DECtape I/O.

```
*L
            DEFINE DECTAPE
            WRITE 3,9
            FAC=1
            DO 1Ø N=1,2Ø
            WRITE 3,2Ø,N,FAC
            EN=N+1
            FAC=EN*FAC
1Ø          CONTINUE
            STOP
2Ø          FORMAT (,E)
9           FORMAT ("FACTORIAL PROGRAM"/)
            END
*
.FORT
*OUT-S:TE
*
*IN-S:TES1
*
↑                   /COMPILATION FINISHED.  CNTRL C WAS THEN TYPED
.FOSL
*IN-S:TE
*
*OPT-S
*OUT-D7:IO           /DECTAPE 7 MUST ALWAYS BE SPECIFIED FOR I/O
*
*IN-
*
*READY
↑                   /PROGRAM WAS EXECUTED BY TYPING CNTRL P
!
.PIP                /CALLED IN PIP TO LIST THE FILE WRITTEN ON TAPE 7
*OPT-A              /BY OUR FORTRAN PROGRAM
*OUT-T:
*
*IN-D7:IO
*FACTORIAL PROGRAM

1           Ø.1ØØØØØE+1
2           Ø.2ØØØØØE+1
3           Ø.6ØØØØØE+1
4           Ø.24ØØØØE+2
5           Ø.12ØØØØE+3
6           Ø.72ØØØØE+3
7           Ø.5Ø4ØØØE+4
8           Ø.4Ø32ØØE+5
9           Ø.36288ØE+6
1Ø          Ø.36288ØE+7
11          Ø.399168E+8
12          Ø.479ØØ1E+9
13          Ø.6227Ø2E+1Ø
14          Ø.871782E+11
15          Ø.13Ø767E+13
16          Ø.2Ø9227E+14
17          Ø.355686E+15
18          Ø.64Ø236E+16
19          Ø.121644E+18
2Ø          Ø.243289E+19
*OPT-
```

PDP-8 FORTRAN-D (DEC-D8-AFA1-PB)

Problem with storage of SAM BLOCK numbers.

In Disk Fortran both the Compiler and the Operating System exist as two part tapes which are stored on the Disk as four unique files. The Compiler Loader, or driver, is saved on the system device under the file name FORT while the main body of the Compiler is saved under the file name .FT. and is located and loaded by FORT whenever necessary. Similarly, the Operating System loader, or driver, is saved under the file name FOSL while the main body of the Op. Sys. is saved under the file name .OS. and it is located and loaded by FOSL whenever necessary.

In order to locate their associated files, FORT and FOSL do a directory look up for the required name (FORT will look up .FT. and FOSL looks up .OS.). In doing this both FORT and FOSL assume that the first SAM BLOCK number is stored in the third word of each DN (directory name) block. The Monitor, however, stores this number in the third word of the first DN block only (block 177).

If .FT. is not one of the first twenty-five (decimal) files saved on the system, its Directory Name entry will not be in the first DN block. When FORT finds the DN entry for .FT. it will assume the third word of the same DN block to contain the block number of the first SAM BLOCK when in truth it does not. FORT will therefore be unable to load .FT. and the compiler will not run. A similar situation is encountered by FOSL if .OS. is not among the first 25_{10} files saved.

To correct this situation both FORT and FOSL should be made to use a constant 2ØØ as the first SAM BLOCK number (which it always is). This is done as follows:

Load the Fortran Compiler Loader (FORT) and change locations:

LOCATION	FROM	TO
1456	1672	1272
1472	7402	Ø2ØØ

Save the Compiler Loader as usual. Load and save the Compiler main body as usual. Load the Fortran Operating System Loader (FOSL) and change:

1Ø56	1672	1272
1Ø72	46Ø2	Ø2ØØ

Save the OP. Sys. Loader as usual. Load and save the OP. SYS. main body as usual.

Both the Compiler and the Operating System will now function properly no matter which DN block they are entered in when saved.

(See the article DISK MONITOR SYSTEM 4 for a complete description of how to patch a system program, also see the Disk Monitor Manual DEC-D8-SDAB-D.)

PDP-8 FORTRAN-D (DEC-D8-AFA2)

Patch to correct two error conditions during compilation time.

The following two error conditions are not properly flagged during compilation time.

1. In a GO TO statement the absence of a comma between the closing parenthesis and the operand may or may not produce an error message, depending on the second character of the variable. If this character is a floating point designator, the error will be flagged as such. But if it is a fixed point designator, no error will be detected.

   The following patch will correct this problem by adding a new syntax error code Ø2.

```
6Ø24    5357            JMP PATCH1

6157    15Ø1    PATCH1,TAD I Z END      /GET NEXT CHARACTER
616Ø    21Ø1            ISZ Z END      /INDEX TO NEXT
6161    1367            TAD MCOMMA     /CHECK FOR
6162    765Ø            SNA CLA        /COMMA
6163    5225            JMP BACK       /O.K., CONTINUE
6164    4454            ERROR          /NOT FOUND
6165    4ØØ2            4ØØ2           /SYNTAX ERROR
6166    5446            JMP I Z NEWL   /GET NEXT LINE
6167    7524    MCOMMA,-254            /COMMA
```

2. If an apostrophe (single quote) accidentally appears in any statement, an error message (usually 14, system overflow) may or may not be generated depending on where the apostrophe appears. The apostrophe at the end of a line signifies a continuation of the statement to the next line. (Maximum number of characters must not exceed 128.) However, if it appears anywhere except before the carriage return at the end of a line it should be flagged as an illegal character (code 12).
   The following patch corrects this problem:

```
Ø277    5365            JMP PATCH

Ø365    1363 PATCH      TAD M215       /CHECK TO SEE IF NEXT
Ø366    765Ø            SNA CLA        /CHAR IS A CARRIAGE RETURN
Ø367    5217            JMP FILB       /YES, O.K., CONTINUE
Ø37Ø    2145            ISZ Z STMT     /NO, INDEX STATEMENT NO.
Ø371    7ØØØ            NOP
Ø372    4454            ERROR          /SINGLE QUOTE IS
Ø373    4Ø12            4Ø12           /ILLEGAL CHARACTER
Ø374    5342            JMP OVFL+4     /IGNORE REST OF STATEMENT
```

PDP-8 FORTRAN-D COMPILER (DEC-D8-AFA2-PB)

Notice of temporary restriction in Subscript Expressions

Subscript expressions which appear in input/output statements, such as:

```
READ  3,1ØØ,     LIST (expression)

WRITE 2,77,      PACK (expression)
```

may only be of the form (V), (C), (V+C or (V-C), where V is a simple integer variable and C is an unsigned interger constant.

Notice that more elaborate subscript expressions may be used outside of I/O statements, permitting the user to redefine the above sequence in a manner such as:

```
L=(expression 1)
K=(expression 2)
READ  3,1ØØ,     LIST (L)
WRITE 2,77,      PACK (K)
```

or even:

```
LL=LIST (expression)
KK=PACK (expression)
READ  3,1ØØ,     LL
WRITE 2,77,      KK
```

PDP-8 8K FORTRAN COMPILER (DEC-08-A2B1-PB)

Problem with labelled FINI statements

There is a problem in the 8K Fortran Compiler with respect to labelled FINI statements. The current version of the compiler (DEC-08-A2B1-PB) will correctly handle a label on a FINI statement if that statement is the object of a DO statement, but not if it is the object of an IF or GO TO statement. In the latter cases, the compiler will not generate a label for the FINI statement and the label will appear as undefined in the SABR symbol table listing. The following patch to the compiler will correct this problem:

FIELD	LOCATION	OLD CONTENTS	NEW CONTENTS
1	2665	1372	Ø572
Ø	Ø572	unused	4576
Ø	Ø573	unused	764Ø
Ø	Ø574	unused	551Ø
Ø	Ø575	unused	4525
Ø	Ø576	unused	5777
Ø	Ø577	unused	6321

LAB-8 ADVANCED AVERAGER

Note on running the Advanced Averager from PS/8

Because PS/8 operates differently than the disk monitor and other monitor systems, it is necessary to change the Advanced Averager saving and loading procedure when running under PS/8. PS/8 does not leave all areas of core intact when a program is loaded. Since the Advanced Averager depends upon blocks of core being untouched between sections, it is necessary to insure the integrity of all core other than that which receives the new program.

The following is presented as a means of preserving core between sections. It is not the only method and user solutions are encouraged. Unless an RK08, 800K word disk is the system device, it is suggested that the user run the Advanced Averager from the disk monitor system.

To run the Advanced Averager from PS/8, save all sections of the program on disk using PIP with the /B option. Load the first section with ABSLDR (/G=600). At the end of this section monitor returns. Now, all the areas of core not used by monitor must be saved and reloaded for the following section.

```
.SAVE  SYS:CORE  0-7577,  1000-17577
```

With ABSLDR, load CORE.SV using the /I option, then load the next section of the Advanced Averager using the /G option.

Follow this core save-reload procedure prior to the loading of each section and all will work properly.

LAB-8 BASIC AVERAGER

Patch to correct problem with control R

Note this patch is not applicable to the LAB8/e

When using a LAB-8 and the programs Basic Averager (DEC-LB-U21B-PB) and Basic Averager Control Tape (DEC-LB-T21A-PB), the user may encounter the following problem. Sampling one channel and doing a single sweep signal average, will produce the proper display. However, adding to it with a control R (↑R) may produce a stair step pattern on the scope instead of the new averaged signal. The following patch eliminates this problem:

```
6617/ 462Ø          JMS  I  .+1
662Ø/ Ø2ØØ            200

 2ØØ/   Ø               Ø
 2Ø1/ 3161          DCA 161
 2Ø2/ 316Ø          DCA 16Ø
 2Ø3/ 6335          XRIN  XRCL
 2Ø4/ 72ØØ             CLA
 2Ø5/ 22ØØ          ISZ  2ØØ
 2Ø6/ 56ØØ          JMP  I  2ØØ
```

LAB-8/e DAQUAN

Tape Problem

Because DAQUAN program tape (DEC-LBOU8ØB-PB) was assembled without the needed floating point package, it will not run. A new tape is being generated, and as soon as possible, the Program Library will be updating all customers who received their machines after September 1, 1971.

LAB-8/e BASIC-RT

Serious problem reported in BASICE.27

Occasionally, when a user types a control C, BASIC will malfunction possibly with no indication of trouble. The conditions necessary to produce this error are time dependent, and therefore may never be noticed. The cause is failure to clear a flag when control C is typed.

To temporarily correct this problem, after BASIC is loaded, change location 7160 in field 0 from a 0776 to a 7756. This will completely solve the problem. However, the "LIST*" command must not be used if this patch is inserted.

If a LIST* command is given, BASIC will not reset the flag and all output will have additional fill characters after the carriage return-line feed.

This problem will be solved in future revisions which will be announced in the newsletter.

Note: Customers should be aware that future versions of BASIC will have different core locations assigned to its values, and thus user overlays probably will no longer function. It may be necessary for them to reassemble their overlays for the new releases of BASIC.

LAB8/e ADVANCED AVERAGER

Patch included in revision

The LAB8/e Advanced Averager program needs the following patches to prevent it from randomly crashing after the compiler phase Section I.

After Section I is loaded:

LOCATION	SHOULD BE
162	ø
175	ø

This has been included in a new revision which is available from the Program Library.

Customers may order this software for the applicable fee:

Paper Tape Binary	DEC-LB-U61B-PB	$5.00
Paper Tape Binary	DEC-LB-U63B-PB1	$5.00
Paper Tape Binary	DEC-LB-U63B-PB2	$5.00
Paper Tape Binary	DEC-LB-U63B-PB3	$5.00
Paper Tape Binary	DEC-LB-U63B-PB4	$5.00

These prices apply to U.S. customers only. All others should consult their DEC field office for applicable charges and for placing their order.

SMALL COMPUTER HANDBOOK

Documentation Addition

The user should be aware that when using a PDP-8/e with KE-8E-EAE the mode-changing instruction SWAB causes a transfer of the AC to the MQ and clears the AC.

This is not documented in the Small Computer Handbook 1972.

LAP6-DIAL ASSEMBLER

Note on LINC mode assembling

At times, the LAP6-DIAL Assembler appears to incorrectly assemble LINC mode instructions. The following is an example of the most common:

```
0000                        *20
0001                                LMODE
0002                                SEGMNT 3
0003        0000  1020              LDA I
0004        0001  6207              B
0005        0002  6206              STC C
0006
0007
0010
0011
0012                                PMODE
0013                                *6200
0014        6200  1606              TAD I C
0015        6201  4215              JMS PRINT
0016
0017
0020
0021
0022        6206  0000  C,          0
0023        6207  6400  B,          ZLIST
0024        6210  0000              0
0025        6211  0000              0
0026        6212  0000              0
0027
0030
0031        6215  0000  PRINT,      0
0032                                *6400
0033        6400  0301  ZLIST,      301
0034        6401  0302              302
0035        6402  0303              303
0036        6403  0000              0
0037        6404  0000              0
0040        6405  0000              0

NO ERRORS

B       6207
C       6206
PRINT   6215
ZLIST   6400
```

In the previous example, notice that B and C have full 12 bit values associated with them (see the chart on page 3-3 of the LAP6-DIAL Manual, DEC-12-SE2D-D). Line 5 equals 62Ø6 rather than 42Ø6 because when STC, which equals $4ØØØ_8$, is OR'ed with C, which equals 62Ø6, the yield will be 62Ø6 rather than the expected 42Ø6.

The problem is caused by the user symbols defined in PMODE, and then referenced in LMODE, retaining their full 12 bit values. Instead of assembling as an STC into relative location 2Ø6 of the particular segment, the assembler yields a JMP to 2Ø6 of that segment. (The assembler cannot strip a symbol down to 10 bits because in line 4, 12 bits are really necessary.)

Depending upon the particular program, a good programming practice to follow is to add an &1777 or an !2ØØØ to the instruction in order to turn on or off the necessary bits as shown in the following example:

```
0000                        *20
0001                                    LMODE
0002                                    SEGMNT 3
0003        0000    1020                LDA I
0004        0001    6207                B
0005        0002    4206                STC C&1777
0006
0007
0010
0011
0012                                    PMODE
0013                                    *6200
0014        6200    1606                TAD I C
0015        6201    4215                JMS PRINT
0016
0017
0020
0021
0022        6206    0000    C,          Ø
0023        6207    6400    B,          ZLIST
0024        6210    0000                Ø
0025        6211    0000                Ø
0026        6212    0000                Ø
0027
0030
0031        6215    0000    PRINT,      Ø
0032                                    *6400
0033        6400    0301    ZLIST,      301
0034        6401    0302                302
0035        6402    0303                303
0036        6403    0000                Ø
0037        6404    0000                Ø
0040        6405    0000                Ø

NO ERRORS
```



```
B       6207
C       6206
PRINT   6215
ZLIST   6400
```

PDP-12 LAP6-DIAL BUILD

Disk initialization/Blocks 47Ø - 477 overlaid on disk

As currently implemented, LAP6-DIAL will copy blocks 3ØØ to 477 from LINCtape to disk whenever the disk is initialized, i.e. LS=Ø7Ø1; RS=731Ø; I/O Preset; DO; Start 2Ø. This will result in block 47Ø to 477, usually reserved for file area, to be overlaid be whatever is contained in blocks 47Ø to 477 on the LINCtape. To correct this problem, make the appropriate patch:

In the source (BUILD DEC-12-ZR5B)

LINE NUMBER	FROM	TO
Ø674	BM2Ø, -2Ø	BM2Ø, -17

On a LAP6-DIAL SYstems tape

TAPE BLOCK	WORD	FROM	TO
31Ø	373	776Ø	7761

Two bootstraps to load LAP6-DIAL-MS directly from the disk

The following two bootstraps should be used to load LAP6-DIAL-MS directly from the disk (RF08 or RK08). Each bootstrap, when assembled, will yield one block of binary that must be moved to block Ø of the respective disk via PDP-12 PIP. The following instructions should be followed to assemble the bootstrap needed:

1. Type up the program via LAP6-DIAL EDITOR

2. →ZE (zero binary work area)

3. →AS (assemble program)

4. →SB (save binary of program)

5. →DX (note where the one block of binary is saved)

6. →PIP* (load PIP and transfer the one block of binary to R1Ø (disk Ø), block Ø)

7. Toggle in the two or four word bootstrap and LAP6-DIAL-MS will start up.

* After the PIP transfer, the user should create an index entry (via PDP-12 TED) in the DIAL index so that DIAL will not save a file over this binary.

The procedure above assumes that the disk has been initialized See LAP6-DIAL Manual, DEC-12-SE2D-D; Appendix A, section A.3.

```
                *20
                        PMODE
                        *20
               /RF08 BOOTSTRAP FOR DIAL-MS
               /
               /COPYRIGHT 1971
               /DIGITAL EQUIPMENT CORPORATION
               /MAYNARD, MASS,  01754
               /
               /7/14/71
               /SGW
               /
               /FIRST, ASSEMBLE THE PROGRAM BELOW AND MOVE THE
               /BINARY TO BLOCK 0 OF UNIT 10 (RF08 DISK),
               /TO PRESERVE BLOCK ZERO, IT MAY BE WISE TO FAKE
               /AN ENTRY IN THE INDEX OF UNIT 10 SHOWING
               /BLOCK 0 AS USED, SO THAT SOMETHING DOESNT
               /GET SAVED THERE AND DESTROY THE BOOTSTRAP.
               /THEN,
               /TO USE:
               /TOGGLE IN THE FOLLOWING:
               /       *20     /PMODE
               /       6603    /READ
               /       5021    /JMP .
               /
               /       *7750
               /       7577    /WORD COUNT-1
               /       7777    /CORE LOC-1
               /
               /SET MODE = PMODE
               /I/O PRESET
               /START 20
               /
               /
0020   6623            DISK            /THIS OVERLAYS THE 6603
0021   5020            JMP     .-1     /AND THIS OVERLAYS THE JMP .
0022   1044            TAD     P6777
0023   3445            DCA I   P7750   /LOAD WORD COUNT-1
0024   1044            TAD     P6777
0025   3446            DCA I   P7751   /AND ADDRESS-1
0026   1041            TAD     P10
0027   6615            DIML            /LOAD EXTENDED ADDRESS (FIELD 1)
0030   1042            TAD     P15     /THIS IS BLOCK 322 IN DISGUISE
0031   6643            DXAL
0032   1043            TAD     P1000   /,,,MORE OF BLOCK 322
0033   6603            6603            /READ
0034   6623            DISK            /SKIP ON DONE
0035   5034            JMP     .-1
0036   6213            CIF CDF 10
0037   5440            JMP I   .+1
0040   7777            7777            /BOOT DIAL NOW
0041   0010    P10,    10
0042   0015    P15,    15
0043   1000    P1000,  1000
0044   6777    P6777,  6777
0045   7750    P7750,  7750
0046   7751    P7751,  7751
                DXAL=6643
                DIML=6615
                DISK=6623
```

```
                *20
                /RK08 BOOTSTRAP FOR DIAL-MS
                /
                /COPYRIGHT 1971
                /DIGITAL EQUIPMENT CORPORATION
                /MAYNARD, MASS.  01754
                /
                /FIRST, ASSEMBLE THE PROGRAM BELOW AND MOVE
                /THE BINARY TO BLOCK 0 OF UNIT 10 (RK08 DISK).
                /IT MAY BE WISE TO FAKE AN ENTRY IN THE INDEX
                /OF UNIT 10 TO PROTECT BLOCK 0 SO THE BOOTSTRAP
                /DOESNT GET DESTROYED.
                /
                /TO USE:
                /TOGGLE IN THE FOLLOWING:
                /       *20     /PMODE
                /       6733    /READ
                /       5021    /JMP .
                /
                /SET PMODE
                /I/O PRESET
                /START 20
                /
                /THE PROGRAM BELOW IS ORIGINED AT *17, BUT IT ACTUALLY
                /GETS LOADED AT *20, WHICH IS WHY ALL THE REFERENCES
                /ARE OFF BY 1.
                /
                        PMODE
                        *17
0017  6745              DSKD            /SKIP DISK DONE (OVERLAYS THE 6733)
0020  5020              JMP     .-1+1   /THIS IS REALLY JMP .-1  (OVERLAYS THE JMP .
0021  1040              TAD     P10+1
0022  6732              DLDC            /FIELD 1
0023  1041              TAD     P6777+1 /STARTING ADDR, -1
0024  6755              DLCA
0025  1042              TAD     P7000+1 /WORD COUNT
0026  6753              DLWC
0027  6742              DCLS
0030  1043              TAD     P322+1
0031  6733              DLDR            /READ BLOCK 322
0032  6745              DSKD
0033  5033              JMP     .-1+1
0034  6213              CIF CDF 10
0035  5437              JMP I   .+1+1   /BOOT DIAL NOW
0036  7777              7777
0037  0010  P10,        10
0040  6777  P6777,      6777
0041  7000  P7000,      7000
0042  0322  P322,       322
                        DSKD=6745
                        DLDC=6732
                        DLCA=6755
                        DLWC=6753
                        DCLS=6742
                        DLDR=6733
```

PDP-12 DIAL-MS

Patch to force Linctape to be the system device

There are a number of occasions when it is desirable that DIAL-MS not use a disk as the system device, even though one is present on the system. Among these instances are the following cases:

1. Using the FPP Assembler on a system with one DF-32 disk: the Assembler requires DIAL-MA, but also requires that if DF-32's are used as the system device, at least two must be present.

2. Using Focal-12 under these same circumstances.

3. Initializing a tape on a system with an inoperable or malfunctioning disk.

4. Starting up DIAL-MS on a system in which the disk must not be overwritten, e.g., in a CL-12 or PS/12 situtaion.

The following patch to the DIAL-MS system tape solves this problem by allowing sense switch Ø to affect the choice of a system device. If SS Ø is in the Ø position, DIAL-MS is initialized in the same manner as it currently is. If SS Ø is in the 1 position at the time of initialization, however, Linctape will be chosen as the system device regardless of what disks are present on the system.

BLOCK	REL. LOCN.	OLD VALUE	NEW VALUE
310	Ø14	Ø	ØØ44Ø
310	Ø15	Ø	6Ø36
310	Ø16	Ø	ØØØ2
310	Ø17	Ø	5766
310	Ø35	ØØ11	6Ø14

PDP-12 EDITOR

Attention PDP-12B ½ (A-D's, but no relays) users.

The LAP6-DIAL V2 EDITOR checks to make sure the relays are available, to tell whether to use the analog potientiometers in the Editor. To get around this, the following patch should be used:

1. Load block 312 into core via Ø7ØØ Ø312; I/O preset: DO
2. Change linc location 4Ø46 from 765Ø to 72ØØ
3. Rewrite back onto tape block 312 via Ø7Ø4

PDP-12 LAP6-DIAL PIP

Patch to allow PIP to save programs on a 1600_8 block LINCtape

To allow PIP (on LINCtape DEC-12-SE2E-UO) to save files on a 1600_8 block LINCtape, make the following changes to the binary:

1st block of the actual binary (starting block in index + 1)

WORD	FROM	TO
113	Ø777	1577

7th block of the actual binary (starting block in index + 7)

WORD	FROM	TO
356	Ø777	1577

Note: With this change the "U" option in auxilliary mode will still copy only the first $1ØØØ_8$ blocks.

The following patch may be used on the PIP source (on LINCtape DEC-12-SEZB-UO, part 1)

PIP Source

LINE NUMBER	FROM	TO
0237	TUPPER=777	TUPPER=1577

Note: With this source change, once PIP is reassembled and saved, the "U" option will copy a full 1600_8 blocks.

NOTE! Please notice the above change from header block to starting block number, when you display the index of a LAP6-DIAL tape.

PDP-12 LAP6-DIAL PIP

Patch to allow PIP's "D" option to copy a 1600_8 block LINCtape

To allow PIP (on LINCtape DEC-12-SE2E-UO) to copy a 1600_8 block tape using the "D" option make the following changes to the binary:

14_8th block of the actual binary (starting block in index $+14_8$)

WORD	FROM	TO
221	7332	1113
222	7Ø1Ø	7ØØ1

The following patch may be used on the PIP(4) source (on LINCtape DEC-12-SEZB-UO, part 1).

LINE NUMBER	FROM	TO
1264	CLA CLL CML RTR	TAD EUPPER
1265	RAR	IAC

NOTE: This patch assumes the LAP6-DIAL PIP 1 patch has also been implemented.

LINKING LOADER (PAPER TAPE VERSION)-DEC-Ø8-A2C3-PB

Failure to initialize low-speed reader on PDP-8/E

The current version of the paper tape LINKING LOADER does not issue a KCC instruction to initialize the low-speed reader. This fact does not effect users with a PDP-8, 8I, or 8L, as the START key sets reader run. However, this is not true on the 8E, and users trying to load via the low-speed reader will not be able to do so. The following patch to the LOADER will correct this problem. It should be made after loading the LOADER.

LOC	OLD CONTENTS	NEW CONTENTS
62ØØ	(72ØØ)	6Ø32

SOURCE PATCH - CHANGE THE 'CLA' AT 62ØØ to A 'KCC'.

KE8E EAE INSTRUCTION TEST 1, MAINDEC-8E-DOLA

Patch

The problem is the Maindec does not check that normalize in "B" mode will clear the AC, if AC=4000 and MQ=0000 prior to issuing a normalize.

The way to correct the problem is to toggle the following changes into memory after loading the program:

LOCATION	FROM	TO:	
4741	7447	5360	/GO to Patch
4760		7431	/"B" Mode
4761		7621	/AC and MQ=O
1762		7330	/AC=4000;MQ=oooo
4763		7411	/ Normalize
4764		7440	/ AC should be O
4765		7402	/ Normalize failed to clear AC
4766		7447	/"A" Mode
4767		5342	/ Exit

KE-8E (EAE) INSTRUCTION TEST 1 MAINDEC-8E-DOLA PATCH

The problem is the "TO BE ADDED" printout that occurs on a "DAD" failure, always indicates zero is being added to the AC and MQ.

The way to correct the problem is to toggle the following changes into memory after loading the program:

LOCATION	FROM	TO
7236	1023	1037
7241	1024	1040

PDP-8E-TELETYPE AND KL8 ASYNCHRONOUS DATA CONTROL TESTS

MAINDEC-8E-D2AB, Patch

The problem is this program cannot test a KL8 jumpered for 2400 Baud transmit and receive.

The patch which can correct this problem is as follows:

1. Toggle into memory the following changes after loading the program:

LOCATION:	FROM:	TO:
0141	7754	7767
0204	4776	5776
0504	N/A	4777
0505	N/A	7630
0506	N/A	7500
0507	N/A	7200
0510	N/A	6600
0511	N/A	5400
0512	N/A	7670
1574	7773	7772
2626	7636	7637

2. Remove G field jumpers and connect G9 and G10 if they exist.

3. If G9 and G10 do not exist, connect E 5 pin 11 to split lug G1 by means of a clip lead and IC clip.

4. Deposit 2400 in location 22. (program can still operate at previous baud rates.)

5. Run program's 0 and 1. Test board at 150/150 2400/2400.

MI8-E BOOTSTRAP DIAGNOSTIC

MAINDEC-8E-D1IA, Patch

This program does not check the possibility that the MI8-E option under test could have affected the memory locations between the bootstrap block of data information and the diagnostic. Insert the following patch to correct the problem.

LOW VERSION		HIGH VERSION	
LOCATION	CONTENTS	LOCATION	CONTENTS
74Ø	537Ø	474Ø	537Ø
77Ø	4341	477Ø	4341
771	5215	4771	5215

MR8-EA Read Only Memory Test

MAINDEC-8E-D1JA

The following patch corrects the problem of data being taken from the wrong address to produce incorrect data in the error typeout.

For the low version (MAINDEC-8E-D1JA-PB1) change the contents of address 1633 to 1775 (TAD ACHECK).

For the high version (MAINDEC-8E-D1JA-PB2) change the contents of address 6433 to 1773 (TAD ACHECK).

DF32 DISCLESS Logic Test, MINIDISC (MAINDEC-08-D5BB)

Patch and revision

Because of the speed at which a PDP-8E or a PDP-12 with ECO EP-00021 and above handles data breaks, the "DATA REQUEST LATE" test in this program always fails. This does not mean that there is a hardware problem, only that it cannot be tested.

The following temporary patch causes the DRL test to be by-passed.

LOCATION	OLD CONTENTS	NEW CONTENTS
2620	7410 (SKP)	7000 (NOP)

This problem has been corrected in a revision of this diagnostic program. Customers may order the following items for the applicable fee.

Binary Paper Tape	MAINDEC-08-D5BC-PB	$5.00
Document	MAINDEC-08-D5BC-D	$5.00

All prices are in U.S. dollars and apply to U.S. customers only. Because currency and import considerations affect the prices in all countries outside the U.S., all other users must check with the nearest Digital office or representative to place their orders.

U.S. customers may send orders with purchase order, check or money order directly to

Software Distribution Center
Digital Equipment Corporation
146 Main Street
Maynard, Massachusetts 01754

AD8E, AM8E A-D CONVERTER & MULTIPLEXER DIAGNOSTIC (MAINDEC-8E-D6BB)

Problems with incorrect tag and switch setting

Two problems have been reported in this diagnostic.

There is an incorrect tag which can be rectified by the following patch.

LOCATION	OLD CONTENTS	NEW CONTENTS
2704	5245	5247

Another problem is that the program fails when SWØ is not set to a one (printout inhibit) when running the Resolution Accuracy Test.

FLOATING POINT INSTRUCTION TEST 2C (MAINDEC-12-DØOB)

Patch to correct time out loop

The following minor correction is needed for a time out loop.

LOCATION	OLD CONTENTS	NEW CONTENTS
4071	1367	4505
4072	3107	1367
4073	4505	3107

PDP-8/E TELETYPE AND KL8 ASYNCHRONOUS DATA CONTROL TESTS (MAINDEC-8E-D2AC)

Patch to correct erroneous characters

If SR2 is set in program 4 to loop the test, erroneous characters are printed out.

To correct this problem, toggle the following change into memory after loading the program.

LOCATION	OLD CONTENTS	NEW CONTENTS
0245	7500 (SMA)	7700 (SMA CLA)

RF08 DISK DATA (MAINDEC-08-D5EB)

Patch to check IOT operation

This program fails to verify that IOT 6603 (DMAR) clears the AC. To correct this problem, make the following changes.

LOCATION	OLD CONTENTS	NEW CONTENTS
3174	7440	SZA
3175	7402	HLT/ERROR
3176	7200	CLA
3177	5756	JMP I READ

PDP-8 MATH ROUTINES (DEC-08-FMCB)

Error in the dividend check routine in the Single Precision Divide Subroutine

A difficulty has been noted in the Single Precision Signed Divide Subroutine (DEC-08-FMCB) of the Math Routine Package. If the high order dividend specified is zero, the low order dividend is treated as a signed number in the check for the fractional quotient. If that part of the dividend is between 4000_8 and 7777_8, it is considered negative and a fractional quotient error results. To correct this problem, the source tape as supplied by the Program Library should be edited as follows and reassembled (changes are underlined):

OLD CODE

```
        JMP CONT
        TAD DIVSOR
        TAD LDIVND
        SPA CLA
        JMP DFRAC
CONT,   CLL
```

NEW CODE

```
        JMP CONT
        CLL
        TAD DIVSOR
        TAD LDIVND
        SNL CLA
        JMP DFRAC
CONT,   CLL
```

PDP-12 MILDRED (DEC-12-FZDA)

Patch to allow reference of units Ø - 77

The current version of MILDRED will only reference units Ø - 17. To allow referencing of units Ø - 77, change the following location:

IN THE SOURCE:

LINE NUMBER	FROM	TO
Ø2Ø7	776Ø	77ØØ

IN THE BINARY:

Change the following word in the second block of the binary (second block = starting block + 1)

WORD	FROM	TO
124	776Ø	77ØØ

PDP-12 MILDRED (DEC-12-FZAD)

Reassembling MILDRED in odd numbered segments

As currently implemented, MILDRED cannot be reassembled in an odd numbered segment (ie segment 1, 5 or 7). To correct this problem, change the following locations:

In the Source:

LINE NUMBER	FROM	TO
Ø177	PPNT1	PPNT1&777
Ø2ØØ	STC PPNT2	STC PPNT2&777

If you are reassembling MILDRED, remember to add in a Segment X (X=Ø to 7) pseudo-op, followed by a *2Ø in the beginning of the source.

PDP-12 MILDRED

Patch to allow MILDRED to address a 1600_8 block LINCtape

To allow MILDRED to create index entries on a 1600_8 block LINCtape, make the following changes to the source and reassemble to generate a new binary.

MILDRED (source) DEC-12-FZDA

LINE NUMBER	FROM	TO
0545	-1000	-1577
1033	7000	6000

OS/8 PIP WITH TD8E

Patch to correct loss of data files

Trying to compress a device onto DTA1 produces a shortened directory on DTA1. In particular, compressing DTA1 onto itself can cause loss of data files. This error only exists on TD8E systems, and only when the tape on DTA1 is a non-system tape.

The correction is:

. GET SYS PIP

. ODT

15276/7620 7640

51352/0177 0171

7600G

. SAVE SYS PIP

PAL-D ASSEMBLER (DEC-D8-ASAC-PB)

Patches to Correct:

1) Use of PAGE pseudo-op
2) Appearance of an extraneous character in the listing when multiple inputs from the disk are used.

The PAGE pseudo-op in PAL-D does not perform correctly when used as follows:

PAGE SYMB

where SYMB is a previously defined symbol. The current location counter is set to the middle of a core page. The following patch will correct this problem:

LOCATION	FROM	TO
0620	0145	1112
0621	1112	0145

If the user desires successive PAGE pseudo-ops with no arguments or a PAGE with no argument at the beginning of a page to be ignored, the following patch should be made:

LOCATION	FROM	TO
0620	0145	1144
0621	1112	0145

This patch also handles the previous problem with PAGE SYMB.

If the user is assembling two disk files with PAL-D and the first file does not end with a PAUSE pseudo-op (it doesn't have to), an extraneous character will be inserted in the assembly listing between the two files. This character does not in any way effect the binary output of the assembly. The following patch will eliminate the character:

LOCATION	FROM	TO
4072	1747	1750
4322	1747	1750

PAL-8

Programming Note on use of conditionals

It is possible to construct useful pseudo-ops such as IFNEG and IFPOS as in the following example:

```
IFNEG expression <statements>  (assemble statements
                                if expression is
                                negative)
```

can be written as:

```
IFNZRO expressions &4000 <statements>
```

while its complement.

IFPOS expression statements, can be implemented by writing:

```
IFZERO expression &4000 <statements>
```

To prevent PAL-8 from printing nonsatisfied conditional assembly statements in the listing, the following solution, employing complementary conditionals, is suggested:

```
IFNDEF  LTAPE <XLIST>
IFDEF   LTAPE<
        HERE
        GOES
        THE
        CODE
IFNDEF LTAPE <XLIST>
```

PDP-8 DISK MONITOR SYSTEM REFERENCE MANUAL (DEC-D8-SDAB-D)

Additional information on copy functions of PIP

The Peripheral Interchange Program of the Disk/DECtape Monitor System (PIP) will transfer files between any of the various devices recognized by the system (i.e. Disk, DECtape, High Speed Paper Tape Reader, High Speed Paper Tape Punch, Teletype Reader, Teletype Punch).

All of the transfer or copy functions use the same conventions to specify the destination (output) and the origin (input) of the file to be copied, and to start the actual transfer once the necessary information has been given. These conventions are essentially:

1) Use the device name followed by the file named to reference any file which exists (input) or is to exist (output) on a file structured mass storage device. For the system device (Disk if present or DECtape) this name is S:
For any non-system DECtape this name is D1:, D2:, D3: etc. depending upon the physical unit setting of the hardware.
The file name consists of up to four ASCII characters.

2) Use only the device name to reference any file which exists or is to exist on a non-file-structured device. For both the high speed paper tape reader and punch this name is R:
For both the teletype reader and punch this name is T:

3) To indicate more than one file (legal only as input to A option; used for merging ASCII files), reference the files by the above rules and use commas as separators.

ex S:FILA, S:FILB, D4:FILN
R:, R:, T:, R:

4) Type CTRL/P (↑P) to start the actual transfer of each file. This is done by depressing the CTRL key while striking the P. PIP will type ↑'s to remind you to type CTRL/P to start transfers if the copy operation is between disk and DECtape, disk and high speed paper tape, DECtape and high speed paper tape, or if it uses the teletype for input but not for output.

If, however, output is to the teletype punch, it is imperative that the user remember the convention and type CTRL/P to start the transfers. PIP cannot type an ↑ as a reminder because it would be punched on the output tape as well as typed. (CTRL/P is not echoed so it won't be punched.)

EXAMPLE:

```
*OPT-A
*OUT-T:↵

*IN-S:FILE↵

A pause occurs here as PIP is waiting for the user
to type CTRL/P

*OPT-A
*OUT-R:↵
*
*IN-S:FILE↵
*↑

A pause occurs here after PIP has reminded the user
it is waiting for CTRL/P.
```

The purpose of the pause is to allow the user time to turn on the punch he wants to use, or to take a DECtape off write lock, or to place a tape in the reader. The copy routine will not proceed until a CTRL/P has been typed.

PDP-8 DF32/RFØ8 PIP (DEC-D8-PDAD and DEC-D8-PDZE)

Information on performance of latest versions of PIP.

PIP can only copy relocatable binary tapes (output of the SABR Assembler) from reader to punch. PIP cannot copy relocatable binaries to or from a mass-storage device (disk or DECtape).

PIP will only copy the first section of a multi-section binary tape when output is to the disk or DECtape, even though the entire tape will go through the reader. A multi-section binary tape is one that has leader/trailer (code 2ØØ) imbedded. The binary copy of such a tape to the mass-storage device will cease once the leader/trailer is encountered, even though PIP will not stop reading the tape until it has received a physical end of tape signal from the reader.

PIP may insert an extraneous character when merging a paper tape with a disk or DECtape file, when the output is going to disk or DECtape. This can happen when the end of the paper tape is jagged and it is interpreted as punches in the tape. It can also be caused by the fact that PIP treats input, character by character when it is from a non file-structured device (teletype or high speed reader) and word by word when it is from a file-structured device (disk or DECtape). Thus, if there is an odd number of characters from the non file-structured device, one half of a word in the output file will be filled with either an @ character or a control character. This can be avoided by first transferring the paper tape to the system device before merging the files.

PDP-8 PIP (DEC-D8-PDAD and DEC-D8-PDZE)

Program halt due to unexpected interrupt

The above noted library versions of PIP will halt at location Ø266 with PC=Ø227, MA=Ø226, MB=7412 and the AC=ØØØØ upon receipt of an unexpected interrupt. An example of this may be an interrupt from a device other than Disk, DECtape, High Speed Reader or Punch, or Keyboard Reader or Punch. Users should be able to eliminate the problem by starting the Monitor at 76ØØ and calling PIP again (start key will clear the flags).

PS/8 MULTI-BLOCK DIRECTORIES

COMPRESSION OF FRAGMENTED DIRECTORY SEGMENTS

When a directory extends beyond the first block, the following situations can occur:

1. If the last entry in a block is an "empty" and you delete the first file in the succeeding block, two consecutive empties will appear in the directory listing.

2. If you delete a file from a directory block, (other than the last block in use) an empty file equal to the length of the file deleted will be created. However, the only way another file may be added to that block is by transferring a file of known length that can be accomodated in an existing empty file. Temporary (newly created files) files are always added at the end of the directory unless you specify a maximum length with the square bracket option. Therefore, "full" directory blocks may have far less than the minimum number of file entries.

3. When you move a file of known or maximum specified length, it will replace the first empty file with a length closest to the desired length. This feature works across blocks beginning with the first block.

Item 3 is an explanation of why files go where they go. Fragmented directory segments are compressed using the IS option in PIP.

This removes all empties by rewriting the files and the directories. This file will have a maximum number of file entries in each directory block up to the last block in use.

8K SABR ASSEMBLER (DEC-08-A2D2-PB)

Problem in Listing PASS with OFF-PAGE Externals

There is a problem in the library version of the SABR assembler (version 16), which can cause SABR not to list a linkage to an off-page external symbol. The binary output of pass 1 contains the correct link-word, but the word is not listed in the pass 2 output. The following patch will correct this problem:

LOCATION	OLD CONTENTS	NEW CONTENTS
3735	(ØØ16)	Ø377
3777	(unused)	Ø4Ø3

8K SABR ASSEMBLER (DEC-08-A2D2 - 16)

TEXT pseudo-op problem

The current version of the 8K SABR Assembler loses track of an internal pointer when it is not in automatic paging mode and it encounters a 'TEXT' pseudo-op that stretches across a page boundary.

The following patch will correct this problem:

LOCATION	FROM	TO
1671	3151	3377
17Ø2	2151	2377
171Ø	1151	1377
172Ø	1151	1377
1722	3151	3377
1743	2151	2377

PS/8 ON TD8E

Patch to correct file operations

In the preliminary release of PS/8 on the TD8E the user is not able to successfully zero the directory or perform a file compression on devices SYS or DSK. When these operations are tried, the number of free blocks remaning on the tape is in error. The problem is that PIP's device length table was not updated to reflect the TD8E device type. The new release of OS/8 does not contain that problem.

To correct the situation, use ODT with PIP as follows:

```
.GET  SYS  PIP
.ODT
13621 /  ØØØØØ 6437
7600G
.SAVE  SYS  PIP
```

This gives SYS and DSK on the TD8E the correct device size.

TSS/8 BASIC V4

Changing the number of columns in the LPRINT command

1) The following patch to BASIC V4 will enable the LPRINT command to handle any number of line printer columns. The maximum number of characters that will appear on a single line is specified at location 1375 in the patch. Any attempt to write more characters than this on a single line will result in wraparound to the next line.

The patch is:

.LOAD BASIC 14157 4ØØ 6226

At Location	Deposit	Comment
11Ø4	4771	
1114	1361	
1361	1517	
1362	22Ø5	
1363	774Ø	
1364	ØØØØ	
1365	1774	
1366	3Ø14	
1367	1375	
137Ø	3774	
1371	1Ø14	
1372	3375	
1373	5516	
1374	251Ø	
1375	nnnn	-number of line printer columns (7660 for 80, 7574 for 132)
4233	5777	
4377	1365	
4771	1411	
4772	16Ø5	
4773	4Ø24	
4774	1717	
4775	4Ø14	
4776	1716	
4777	Ø7ØØ	

THEN:

.SAVE BASIC 14157 4ØØ 6226

TSS-8

Disk/Core Redefinitions

Changes in the number of disks or number of core fields on a Time-Shared 8 system do not require reassembly of the entire Monitor. Following is the information necessary to make the appropriate changes.

Users who are not familiar with the DISKLOOK section of INIT that starts at location 0200 should read Chapter 4 of the Time-Sharing System Manager's Guide. The locations that are listed here refer to Version 22B of the Time-Shared 8 Monitor. Locations in earlier versions may differ slightly from these.

To change the number of disks, either RF/RS08 or DF32:

loc. 23527 becomes -DSKSIZ
loc. 23742 becomes DSKSIZ
where DSKSIZ = (# of disks)x100(octal)

Having made these two changes it is necessary to "REFRESH the disk" using INIT. This, of course, requires reloading all of the files.

Although it is not really necessary to change the Monitor when adding more core, for consistency here are the changes that can be made. Additional core fields can be utilized by correctly answering the INIT question:

OF USER FIELDS?

This number is two less than the total number of fields in the system (8K resident Monitor).

To correct the Monitor for added core fields:

loc. 25400 becomes CORMEM
If 680-type teletype interfacing,
loc. 33437 becomes CIF CDF + CORMEM
loc. 33444 becomes COREXT
If not 680 interfacing,
loc. 33430 becomes CIF CDF + CORMEM
loc. 33435 becomes COREXT
Where CORMEM = (highest field #)x10(octal)
and COREXT = CORMEM if RF/RS08
= 200 + CORMEM if DF32

TSS/8 V22B

Patch to correct LPT/DTA interference

The patch below removes the DTSF from the LPT/PTP driver whenever the DTSF is removed from the interrupt chain and inserts it whenever the DTSF is inserted in the chain. This prevents the LPT driver from dispatching to the DTA routine when the DTA has been disabled by a select error or other causes.

Unit Ø should be selected when the system is started unless the ECO for DTCA has been installed.

The following patch should be made to TS8II.

LOCATION	OLD CONTENTS	NEW CONTENTS	SOURCE
125Ø	3437	4776	JMS I PATCHM
1376	ØØØØ	2156	PATCHM, PATCH
1651	3437	4775	JMS I PATCHL
1712	3437	4775	JMS I PATCHL
1775	ØØØØ	2156	PATCHL, PATCH
2156	ØØØØ	ØØØØ	PATCH, Ø
	ØØØØ	3365	DCA SAVE
	ØØØØ	1365	TAD SAVE
	ØØØØ	3437	DCA I Z 37
	ØØØØ	1365	TAD SAVE
	ØØØØ	3766	DCA I LPTDTI
	ØØØØ	5756	JMP I PATCH
	ØØØØ	Ø	SAVE, Ø
	ØØØØ	7535	LPTDTI, 7535

PDP-12 LAP6-DIAL-MS User's Manual

Correction for Add Binary command

There is an error in the LAP6-DIAL-MS User's Manual as to the actual relocation factor for the binary. The formula on page 4-6 should read:

[(FIELD * 10000 + ADDRESS) - (ORG & 17400)]

The following are a few examples of how this command works:

Lowest orgin in program	Specified Address	Relocated To
* 250	500	750
* 500	500	600
* 1200	4000	4200
* 6300	4000	4300

PDP-12 AIPOS USER'S MANUAL (DEC-12-SQ1A)

Documentation correction

Page 4-1, Section 4.3 of the AIPOS User's Manual (DEC-12-SQ1A-D) states that the L R command to FOCAL-12 will cause a return to JDB Control. The is an error. The R must be followed by a comma (,). Therefore, the format should be:

L R, ↵

PDP-12 MASH USER'S MANUAL

Additional notes on Automatic Mode

The following information on the automatic mode should be inserted in the MASH User's Manual (DEC-12-SQ2A-D), as the third paragraph on page 16.

If G is typed again, automatic mode initiates a series of scans that last until the chromatograph run is complete. Pressing one of the right switches on the console terminates scanning and returns the program to the TIC ys. Time plot.

PDP-12 LAP6-DIAL EDITOR

Note on correct usage of the LAP6-DIAL EDITOR

While working with LAP6-DIAL Editor, the user may experience problems if he encounters the following situation:

4Ø36 4Ø37	/THIS IS THE 1ØØth BLOCK /THIS IS THE LAST LINE	BLOCK 1ØØ
4Ø4Ø	/THIS IS THE FIRST LINE IN BLOCK 1Ø1	BLOCK 1Ø1

Assume there are four spaces after the 1 in 1Ø1. By moving the cursor back using knob 3 to anywhere on the line and pressing a return, line 4Ø4Ø up to the cursor will be stored (written) onto block 47Ø.

The same procedure will work if you are at line 7777, except the next thing viewed on the scope is line 1 and the entire working area is destroyed.

Basically, the above example states the following:
DO NOT turn the cursor back and strike a carriage return in the middle of a line whenever you feel you have a 1ØØ block source program in the working area (or you are at line 7777).

This problem has been corrected in the LAP6-DIAL-MS Version 2. However, it will not be corrected in the LAP6-DIAL-V2 System Editor that goes out on that same tape (DEC-12-SE2A-UO).

PDP-8 Hardware design note for the DF32 disk

Programming note concerning the DEAL instruction

It is now possible for bit Ø of the AC to be set by the photocell sync mark when using DF32 Disk instruction, DEAL (6615). If the user expects the AC to be unchanged, this could effect his program.

Therefore, the user should not write his program such that it depends on the contents of the AC remaining unchanged after a DEAL instruction. Page 2-4 of the DF32 Disk File and Control Instruction Manual specifies only the contents of AC bits 1-8 and claims only these bits will remain unchanged. If the information in the AC is to be used after a DEAL, bits Ø and 9-11 should first be masked out.

PDP-12 PROGRAMMING NOTE

Software Solution to a DJR Problem

A problem occurs when returning to a main program after executing a subroutine in another intruction field. The problem occurs when the DJR is used in the subroutine to save the contents of location Ø of the subroutine's instruction field.

The following text describes the problem and gives two ways of programming around it.

In PDP-12 (L Mode) programming the JMP Ø (zero) instruction, when used in a subroutine, returns the program counter to the calling program because location Ø of the current instruction field (IF) will contain the return jump instruction. If the subroutine is in a different IF than the calling program then the instruction sequence LIF ( ), JMP (subroutine) causes the return jump instruction to be placed in location Ø of the instruction field containing the subroutine. Program control returns to the calling program in a direct, convenient way by the LIF ( ) and JMP Ø instructions at the end of the subroutine. Normally, a JMP Ø instruction that follows the LIF ( ) instruction places the program counter at location Ø of the current (subroutine) IF instead of in the new (calling program) IF.

If the subroutine contains JMP commands then the contents of location Ø are preserved by using the DJR instruction before each JMP instruction. However, if, by the circumstances of the subroutine, a DJR is not cleared (i.e. by a JMP (≠Ø) before the LIF ( ) JMP Ø sequence, then the program counter goes to loation Ø in the IF of the calling program instead of the IF of the subroutine. That is, the DJR instruction not only inhibits the changing of location Ø of the same IF but it also causes the program counter to go to location Ø of the new IF when the JMP Ø is executed. Although this return to location Ø in the calling program's IF could be utilized, it is not consistent with the procedure of saving the return jump in location Ø of the subroutine's IF and then protecting that return jump with the DJR.

Two procedures are presented to circumvent the problem of returning to the wrong IF. Note in procedure 1 that extra instructions, DJR and JMP .+1, must be inserted in the subroutine, and in procedure 2 the instructions JMP .+3, LIF 1 and JMP BSUB must be added to the calling program so that the program counter will return to the proper instruction after the subroutine is executed.

Procedure 1

Storage Location	Calling Program (IF 2)
4000	0000 Contents are not changed by
.	JMP SUB
.	.
.	.
4060	LIF 1
4061	JMP SUB
4062	ADA (Program control returns here after executing the subroutine)
4063	.
.	.
.	.
.	.

Storage Location	Subroutine (IF 1)
2000	JMP 0062 (Return JMP placed here after execution of 4061)
.	.
.	.
.	.
2120	SUB, STA
2121	0
.	.
.	.
.	.
2140	DJR
.	.
.	.
.	.
2150	APO
2151	JMP .+4
.	.
.	.
.	.
2167	DJR (If previous JMP is not
2170	JMP .+1 executed then this DJR
2171	LIF 2 and JMP clears the DJ
2172	JMP Ø and saves the contents of 2000) (Program counter goes to 2000, then the IF is changed to 2 and the program counter goes to location 4062)

Procedure 2

Storage Location		Calling Program (IF 2)	
4000		JMP 062	(Return jump after executing JMP ASUB)
.		.	
.		.	
.		.	
4060		.	
4061		JMP ASUB	
4062		ADA	(Program control returns here after executing the subroutine and the jump at location 4000)
.		.	
.		.	
.		.	
.		.	
.		.	
4111		JMP .+3	
4112	ASUB,	LIF 1	
4113		JMP BSUB	
4114		.	
.		.	
.		.	
.		.	

Storage Location		Subroutine (IF 1)	
2000		JMP 114	(Return jump after executing JMP BSUB)
.		.	
.		.	
.		.	
2120	BSUB,	STA	
2121		0	
.		.	
.		.	
.		.	
2140			
.		.	(DJR is not required. The contents of location 2000 are not used to return to the calling program)
.		.	
.		.	
.		.	
		.	

```
2150        APO
2151        JMP  .+4
  .          .
  .          .
  .          .
2170        DJR      (This DJR will cause the
                      program counter to go to
                      4000 at the execution of
                      the JMPØ instruction at
                      2172)
2171        LIF2
2172        JMPØ
```

TC-58 Hardware Information

The following items may be informative to users with TC-58's as they are not included in the documentation.

a. To change the data field, execute MTGO with the field in bits 6-8 of the accumulator. The accumulator is not cleared.

   For example:

```
TAD FIELD
MTGO
CLA
```

b. A bad tape is detected in all modes except Rewind, Write end of file (EOF), or Write.

c. When Write EOF is completed, the EOF bit is set. Prior to this, EOF marks are not detected.

RTPS FORTRAN IV (FORRTS.FT version 4)

Patch to correct logical input field

It is possible for a logical input field to be incorrectly filled if T or F is not the first character read. Use FPIP to effect the following one location patch to correct this problem.

.R EPIC	Call EPIC
* FORRTS.FT ←/1$	Load FORRTS for editing ($ is altmode)
R,24	Read block 24
O,335	Open location 335
5316/5317	Change 5316 to 5317
W	Write the block
E	Exit
*↑C	Return to OS/8

RTPS FORTRAN IV (DEC-08-LRTDA) (F4.SV version 1.06)

Problem with arithmetic functions

When using arithmetic functions, a problem with temporary storage occurs if an arithmetic operation of more than 1 step is performed before using the arithmetic function on a calculation. For example,

```
ROUT(A) = A*2+SQRT(A)

V =  (3.+C) /ROUT (C)
```

The result of the calculation of 3.+C is destroyed during the calculation of ROUT(C), so the value of V is incorrect.

This problem can be avoided by using an external function instead of an arithmetic function. (Another means would be to use only 1 step arithmetic calculations after the arithmetic function description, but this path is not recommended.) The above calculation will execute successfully if ROUT is described in an external function as:

```
FUNCTION ROUT(A)

ROUT = A*2+SQRT(A)

END
```

RTPS FORTRAN IV (F4.SV version 1.Ø6)

Problem with DO loops in I/O statements

An error occurs if a subscripted variable is the upper limit in a DO loop in an I/O statement. For example:

```
WRITE (4,100) (PAR(I),I=1,4), (DATA(I),I= 1, PAR(1)
```

This problem will be corrected in the next release of FORTRAN. For the current version it is possible to code the program so a subscripted variable is not the upper limit in an I/O statement. For example, J=PAR(1)

```
WRITE(4,100)(PAR(I), I=1,4) (DATA(I),I=1,J)
```

RTPS FORTRAN IV (DEC-08-LRTDA)

Programming Note

The following details of the RTPS FORTRAN IV system may be of interest to some users.

FORTRAN does not add a line feed after each carriage return when transferring formatted data to a mass storage device. Also, null characters in a Hollerith field are currently treated as Ø. More preferable handling would be for the character to be 40. Both of these conditions will be corrected in the next release.

Users should also be aware that, due to the nature of binary numbers, some numbers cannot be represented internally as an exact value. This is most observable after a propogating-type calculation where a result of Ø is anticipated and a value such as $9x10^{-25}$ is returned.