PART 15

THE DOS/BATCH FILE DUMP PROGRAM

FILDMP

PART 15

CHAPTER 1

INTRODUCTION TO FILDMP

1.1 INTRODUCTION

File Dump (FILDMP) is a utility program for use with the PDP-11 Disk Operating System (DOS/BATCH). It can be used to enhance the debugging of programs by providing a printed copy of the contents of all or specific blocks of a file for visual inspection.

Printout of a file or block(s) of data can be directed from an input device to any device capable of ASCII output.

FILDMP takes its input either as filenames or mass storage block numbers, and outputs a dump in various formats. The printed dump is in a readable format. FILDMP is device independent in that the output can be stored on disk or tape for later printing.

FILDMP command strings are interpreted by the DOS/BATCH Command String Interpreter (CSI), as explained in Chapter 3-6. For example, the general format of a FILDMP command string is:

```
output file specification(s)<input file specification
```

or

```
outdev:filename.ext/switch<indev:filename.ext/switch
```

The output file specification on which the data is to appear is usually KB: (terminal) or LP: (line printer), but can be a file on any mass storage device. The /switch represents FILDMP's switch options. The input device, indev:, is the device on which the file is stored, the DOS/BATCH system device is assumed if no input device is specified. Note that all keyboard command strings are terminated with the RETURN key, which is non-printing.

There may be any number of output file specifications (or none, in which case no output is generated). However, there must be exactly one input file specification. The entire command string must not be more than 72 decimal characters, the length of a teleprinter line.

1.2 RUNNING FILDMP

FILDMP is loaded as a DOS/BATCH system program (see the DOS/BATCH System Manager's Guide). It can then be called into core and executed with the DOS/BATCH Monitor RUN command. For example:

```
$RUN FILDMP
FILDMP Vxxx
#
```

FILDMP identifies itself and prints #, indicating its readiness to accept a user command string from the teleprinter keyboard.

PART 15

CHAPTER 2

SWITCHES

FILDMP operations are controlled with switches in the command string. Switches are expressed using a slash and two letters, as shown below. There are two types of switches: input and output.

2.1 INPUT SWITCHES

Input switches are used to:

1. Specify the format in which data is to be read.
2. Determine the mass storage block numbers on which a file is stored.
3. Specify the block or group of blocks to be dumped.

Input switches can appear only in the input field of the command string, i.e., to the right of the < symbol. They are:

/BL	read specified blocks of mass storage. /BL requires at least one value (block number), and will accept at most two values. The syntax for the /BL switch is: /BL:n or /BL:n:m where n and m are octal block numbers.
/CH	causes FILDMP to obtain the numbers of the blocks which are allocated to the specified file. The input device must be a directory-structured device.
/FA	read the input file in formatted ASCII mode.
/FB	read the input file in formatted binary mode.

FILDMP can read data in any of three modes:

Unformatted binary	default mode
Formatted binary	/FB
Formatted ASCII	/FA

When no input switch is specified, data is read in unformatted binary mode. Unformatted binary mode is assumed with the /BL switch

Only one type of input switch can appear in a command string. For example:

```
DT:FILE.EXT/FA
```

is legal, but

```
DT:FILE.EXT/FA/FB
```

is not.

The /BL switch can appear more than once in a command string. For example:

```
DT:/BL:1/BL:7:13/BL:22
```

is legal, but

```
DT:/BL:23/FA
```

is not because only one type of input switch can appear in a command string. In the last example, block 23 will be dumped as directed followed by an error message when /FA is encountered.

2.2 OUTPUT SWITCHES

Output switches are used to specify the format in which the data is to be dumped. They should appear only in the output field of the command string, i.e., to the left of the < symbol. They are:

/AS	read successive bytes of the input file or mass storage block and output each byte as if it were a single ASCII character. The ASCII character set which FILDMP considers printable is (octal) 40 through 137 and 240 through 337. Any value outside these ranges is printed as if it were 137, a special printing character; for example, a left arrow or a heart-shaped character.
/BY	output the input file or mass storage block(s) as a sequence of octal bytes.
/OC	output the input file or mass storage block(s) as a sequence of octal words.
/RA	read successive words of the input file or mass storage block(s), consider each word as a three-character, packed Radix-50 entity, and unpack and output it as such.

When no output switch is specified, the /OC switch is assumed.

Output switches can be used to direct FILDMP to perform certain modes of translation, e.g., a binary file can be dumped in ASCII. However, the /CH switch overrides any output switch and causes output to appear in octal words.

Multiple output file specifications are allowed and, in fact, are common. For example, to dump FILE.EXT (a file written in formatted ASCII mode) in octal words and ASCII characters, use the following command string:

```
LP:/OC,LP:/AS<FILE.EXT/FA
```

where the line printer is the output device and FILE.EXT is on the system disk. In the above example, FILE.EXT is read in formatted ASCII mode and dumped in octal words, and then read in formatted ASCII mode and dumped as ASCII characters. If, for example, FILE.EXT were on paper tape, it would be necessary to reload the paper tape prior to generating the second output. The command string would be:

```
LP:/OC,LP:/AS<PR:/FA
```

The command string:

```
LP:/BY,KB:/RA<DF:/BL:3/BL:17:21
```

directs FILDMP to dump disk (DF:) blocks 3, 17, 20, and 21 on the line printer in octal byte format, and then on the teleprinter in unpacked Radix-5Ø format.

2.3 OUTPUT FORMATS

The output or printed format of the dump differs slightly, depending on the switch used. If no input switch (implied unformatted binary mode) or the /BL switch is specified, the leftmost column of the output is the byte count of the file or mass storage block. If the /FA or /FB switch is specified, the leftmost column of the output is the line number of the file. The physical output line which begins with the line number contains the status and mode bytes and the byte count word as well.

Read errors are indicated by an E between the line number and the status byte. The status byte gives detailed information concerning the error. The E error message appears only on dumps where the input is read in a formatted mode.

If /CH appears in the input field, the output is the block numbers, in sequential order, occupied by the file. No byte count or line count appears.

The output of FILDMP contains a form of identifier. If /BL appears, FILDMP prints the device name and block number (in octal) prior to dumping any given block. Otherwise, the input filename and extension, as specified in the command input, will appear, followed by:

(ASCII)	if the /AS switch is used.
(BYTES)	if the /BY switch is used.
(CHAIN)	if the /CH switch is used.
(OCTAL)	if the /OC switch is used or assumed.
(RAD5Ø)	if the /RA switch is used.

If the input file were read as a file (i.e., /FA, /FB, or no switch), FILDMP terminates its output with:

```
END OF FILE
```

The END OF FILE message does not appear on those dumps which use the /CH or /BL switch.

2.4 DUMPING ENTIRE FILES

Unless the /BL switch (see Chapter) is specified, FILDMP dumps the entire file indicated. When FILDMP encounters an end-of-file (EOF), it prints

```
END OF FILE
```

closes and releases all files, and then prints # to indicate readiness for another command string.

2.5 DETERMINING FILE BLOCKS, /CH

The /CH switch is helpful in determining the block size of a program and in pinpointing certain blocks of data for future referencing with other FILDMP operations. For example:

```
#KB:<DEMO/CH
  DEMO   (CHAIN)
       ØØ2252   ØØ2262
#KB:<SIZE/CH
  SIZE   (CHAIN)
       ØØ17ØØ   ØØ17Ø5   ØØ1712   ØØ1717   ...
       ØØ175Ø   ØØ1755   ØØ1762   ...
       ØØ2Ø2Ø
#
```

DEMO is a file which is stored in the two blocks numbered ØØ2252 and ØØ2262. SIZE is a file which is stored in the 17 blocks numbered as shown.

When using the /CH option, if FILDMP cannot find the input file, then the error message S2Ø6 (no input file) is printed, followed by #. (In no other case does FILDMP attempt to predetermine the existence of its input file, i.e., if the input file is not found, FØ12 results.)

2.6 DUMPING BLOCKS OF DATA, /BL

The block switch, /BL, is used to indicate the dumping of a specific block or group of contiguous blocks. More than one /BL switch can be specified in a command string. Each /BL switch requires at least one and at most two arguments (block numbers), and each argument is preceded by a colon. For example, the following command string dumps the contents of block 2252 in ASCII characters:

```
#LP:/AS<DF:/BL:2252
```

The following command string dumps blocks 17 through 43 in octal words:

```
#LP:/OC<DT1:/BL:17:43
```

The following command string dumps blocks 15 through 21, block 32, and blocks 113 through 121 in octal bytes:

```
#LP:/BY<DK:/BL:15:21/BL:32/BL:113:121
```

The /BL switch reads input in unformatted binary only.

The following command string dumps blocks 7Ø and 1ØØ in octal and then in Radix-5Ø.

```
#LP:/OC,LP:/RA<DT:/BL:7Ø/BL:1ØØ
```

2.7 DUMPING RADIX-5Ø FORMATTED DATA, /RA

The /RA switch can be used to dump Radix-5Ø formatted data in ASCII characters. The /RA switch causes FILDMP to unpack the data (three ASCII characters are packed into one word).

This switch can be useful when "looking" at the data stored in Radix-5Ø format, e.g., internal directories, etc.

The /RA switch can be used alone or with the /FA, /FB, or /BL switch, For example:

```
#LP:/RA</BL:2:3
```

FILDMP would dump blocks 2 and 3 of the DOS/BATCH system device in unpacked Radix-5Ø format.

PART 15

CHAPTER 3

SAMPLE OUTPUT

The following example is not intended to be a practical example of the uses of FILDMP. Rather it is intended to show as many examples of the FILDMP options, output formats, and error notations as possible.

EXAMPLE 1

The FORTRAN source program RAD5Ø.F4, listed below, writes 3Ø records of 12 words each containing the characters A-Z and Ø-9 in packed RADIX-5Ø format. The output is to a file named FORØØ1.DAT on the system device. The RADIX-5Ø packing procedure is described in Chapter 7-12. The CALL SETERR requests that the overflow into bit 15 (the sign bit) be ignored while the RADIX-5Ø characters are being packed.

```
      DIMENSION IN(36)
      DIMENSION IOUT(12)
      DEFINE FILE 1(30,12,U,IERR)
      CALL SETERR(3,-1)
      IFIF=40
      DO 5 J=1,26
5     IN(J)=J
      DO 10 J=27,36
10    IN(J)=J+3
      DO 15 J=1,12
      I=(J-1)*3+1
15    IOUT(J)=((IN(I)*IFIF)+IN(I+1))*IFIF+IN(I+2)
      DO 20 J=1,30
      WRITE(1'J) (IOUT(I),I=1,12)
20    CONTINUE
      CALL EXIT
      END
```

EXAMPLE 2

The FILDMP command string is used to get an ASCII dump of the source file RAD5Ø.F4. The resulting output follows.

```
#LP:/AS<RAD50.F4

  RAD50 .F4    (ASCII)

0000000 ^D      IM      EN      SI      ON       I       N(      36
0000020 )^      ^^      DT      ME      NS      IO       N
0000040 UT      (1      2)      ^^      ^D      EF
0000060 FI      LE       1      (3      0.
0000100 ER      R)      ^^      ^C
0000120 RR      (3      ,-
0000140 40      ^^
0000160 ^^
0000200                         .
                                .
                                .

                                                                T(
                                                        FI      F)
                                                IF      IF      +I
                                        ^D      0       20       J
                                ^^      WR      IT      E(      1'
                        IO      UT      (I      ).      I=      1.
                )^      ^2      ^^      CO      NT      IN      UE
        ^^      ^C      AI      L       EX      IT      ^^      ^E
0000460 ND      ^^

    END OF FILE
```

EXAMPLE 3

The FILDMP command string shown below is used to get a byte dump of the source file. The resulting output follows.

```
#LP:/BY<RAD50.F4
  RAD50 .F4    (BYTES)

0000000 011 104 111 115 105 116 123 111 117 116 040 111 116 050 063 066
0000020 051 015 012 011 104 111 115 105 116 123 111 117 116 040
0000040 125 124 050 061 062 051 015 012 011 104 105 106
0000060 106 111 114 105 040 061 050 063 060 054
0000100 105 122 122 051 015 012 011 103
0000120 122 122 050 063 054 055
0000140 064 060 015 012
0000160 015 012
0000200                                 .
                                        .
                                        .

                                                                124 050
                                                        106 111 106 051
                                                111 106 111 106 053 111
                                        011 104 117 040 062 060 040 112
                                012 011 127 122 111 124 105 050 061 047
                        111 117 125 124 050 111 051 054 111 075 061 054
                051 015 012 062 060 011 103 117 116 124 111 116 125 105
        015 012 011 103 101 114 114 040 105 130 111 124 015 012 011 105
0000460 116 104 015 012

    END OF FILE
```

EXAMPLE 4

The FILDMP command string shown below is used to get an octal dump of the source file. The resulting output follows.

```
#LP:/OC<RAD50.F4
  RAD50 .F4    (OCTAL)

0000000 042011  046511  047105  044523  047117  044440  024116  033063
0000020 006451  004412  044504  042515  051516  047511
0000040 052125  030450  024462  005015
0000060 044506  042514
0000100
```

```
                                                  024105  023461
                                  044450  026051  036511  026061
                          031012  004460  047503  052116  047111  042525
        005015  041411    046101  020114  054105  052111  005015  042411
0000460 042116  005015
```

END OF FILE

EXAMPLE 5

After the source file is compiled to an object file, RAD5Ø.OBJ, the numbers of the physical blocks on the system device that contain the file are determined by use of the following FILDMP command:

```
#LP:<RAD50.OBJ/CH
RAD50 .OBJ  (CHAIN)

      000352  000360  000361  000362
```

EXAMPLE 6

To dump selected blocks of the object file in octal format (the default output format) the following FILDMP command is used:

```
#LP:<RAD50.OBJ/BL:352/BL:361
  DK     000352  (OCTAL)

0000000 000360  000001  000056  000001  050561  055740  000000  000000
0000020 127401  007624  000410  000000  021411  076400
0000040 073634  021042  002100  000000
0000060 000235  000001
0000100
                                  .
                                  .
                                  .
                                                  001010  000174
                                          125517  047000  004002  125361
                          125675  065100  013002  124545  023364  014001
        001002  015001  001004  016001  001006  017001  001010  000123
0000760 000001  000032  000003  000034  000000  000004  000774  000000


  DK     000361  (OCTAL)

0000000 000362  000001  000030  000003  000242  000000  000014  000000
0000020 001060  000470  000000  001026  000000  000263  000410  000000
0000040 000004  002002  125675  065100  001060  001006  000400  000000
0000060 001060  006001  000056  000004  002002  125523  031000  004002
0000100 125067  034100  035002  124626  014657  006001  001002  007001
0000760 001060  010001  001006  011001  000400  012002  125067  022600
```

EXAMPLE 7

The data file containing the packed RADIX-5Ø characters, FORØØ1.DAT, may be dumped in unpacked RADIX-5Ø format by using the following FILDMP command string. The resulting output follows.

```
#LP:/RA<FOR001.DAT
  FOR001.DAT  (RAD50)

0000000 ABC     DEF     GHI     JKL     MNO     PQR     STU     VWX
0000020   0       3       6       9     ABC     DEF     GHI     JKL
0000040 MNO     PQR     STU     VWX       0       3       6
0000060 ABC     DEF     GHI     JKL     MNO     PQR
0000100   0       3       6       9     ABC
0000120 MNO     PQR     STU     VWX
0000140 ABC     DEF     GHI
0000160   0       3
0000200 MNO
0000220
    .
    .
    .
0001700
0001720
0001740
0001760

    END OF FILE
```

EXAMPLE 8

To dump the data file in octal (to see the packed RADIX-50 format) the FILDMP command shown below is used. The associated output follows.

```
#LP:/OC<FOR001.DAT
  FOR001.DAT  (OCTAL)

000000 003223 014716 026411 040104 051577 063272 074765 000030
000020 000036 000041 000044 000047 003223 014716 026411 040104
000040 051577 063272 074765 000030 000036 000041 000044 000047
000060 003223 014716 026411 040104 051577 063272 074765 000030
000100 000036 000041 000044 000047 003223 014716 026411 040104
000120 051577 063272 074765 000030 000036 000041
000140 003223 014716 026411 040104
000160 000036 000041
000200
                                                  000000 000000
                                    000000 000000 000000 000000
                              000000 000000 000000 000000 000000
                 000000 000000 000000 000000 000000 000000 000000
001700 000000 000000 000000 000000 000000 000000 000000 000000
001720 000000 000000 000000 000000 000000 000000 000000 000000
001740 000000 000000 000000 000000 000000 000000 000000 000000
001760 000000 000000 000000 000000 000000 000000 000000 000000

   END OF FILE
```

EXAMPLE 9

If the data file is mistakenly specified as being in formatted binary (the file is actually in unformatted binary, the FILDMP default input file format), FILDMP encounters errors as it tries to read the input file. An example of this FILDMP command string and the resulting output are shown below.

```
#LP:<FOR001.DAT/FB
  FOR001.DAT  (OCTAL)

000000 F004 001 000000
000001 F004 001 000000
000002 F004 001 000000
000003 F004 001 000000
000004 F004 001 000000
000005 F004 001 000000
000006 F004 001 000000
000007 F004 001 000000
000010 F004 001 000000
                .
                .
                .
000257 F004 001 000000
000260 F004 001 000000
000261 F004 001 000000
000262 F004 001 000000
000263 F004 001 000000

    END OF FILE
```