UNITED STATES

DEPARTMENT OF THE INTERIOR

GEOLOGICAL SURVEY

Office of Earthquake Studies
345 Middlefield Road
Menlo Park, CA 94025

DIGITIZING FIVE-DAY-RECORDER TAPES ON THE ECLIPSE AND PDP-11/70 UNIX SYSTEMS

Open File Report 80-610

by

Paul Reasenberg
Jeffrey Hobson

UNITED STATES
DEPARTMENT OF THE INTERIOR
GEOLOGICAL SURVEY

To: Purchasers of Open-File Report 80-610

From: P. Reasenberg, (author)

Subject: Omitted pages

Pages 24-27 have been omitted from the report because they are not sufficiently readable. This omission does not degrade the value of the report because they contain only examples of computer dialogs.

Introduction

Between 1963 and 1965, the Office of Earthquake Research (then called the National Center for Earthquake Reasearch) developed a fleet of 20 portable instrumentation recorders for seismic studies. These recorders were built with Precision Instrument, Inc. tape drives and custom electronics, and they recorded for 10.6 days on a 7200 ft. reel of tape. The first major use of the fleet was at Parkfield, CA immediately following the 1966 earthquake. Since then, although the system has undergone electronic and mechanical modifications, it remains nearly intact. (Nineteen recorders are still in use, one having been stolen in the field.) In the early 1970's the fleet was modified to record at a speed of 5.3 days per reel of tape. The first application of the 5-day system was to record refraction calibration shots at Oroville, Ca. in the summer of 1975. The system has also been used in aftershock studies, as a teleseismic array, and as a seismic noise array.

In the past few years, two avenues have been available for retreiving 5-day-tape data from the recorded analog tapes. In either route, seismic data and IRIG and WWVB timing signals from the 5-day tapes are played back through a set of analog electronic components known collectively as the 5-day-tape playback system (Eaton, 1978). This system "provides for playback of 6 data channels and 2 time channels (multiplexed onto a single tape track along with the compensation reference signal) at playback speeds of 15/16 ips, (which is 5 times recording speed) or 3 3/4 ips, (which is 20 times recording speed). Both capstan

compensation and subtractive compensation...can be applied in any combination."(1) In the past, these eight channels could either be recorded graphically on a Siemans Oscillomink strip chart recorder, or sampled and digitized by a CDC-1700 computer with A/D converter.

In December, 1979, the CDC-1700 computer was turned off and retired from service. At the same time, some changes were made in the arrangement and interconnection of the playback system used for 5-day-tapes. As of this writing, the 5-day-tape playback system still allows either Oscillomink chart output or computer sampling and digitizing of the data. Now, however, the computer route uses the office's Eclipse minicomputer and PDP-11/70 UNIX system.

This report describes the route by which computer sampling, digitizing and archieving of 5-day-tape data are done on the Eclipse and UNIX systems. Four computer programs that run on the Eclipse, and one program running on UNIX are briefly described. For details of the functioning and structure of the software, the reader is referred to the source listings in the Appendix. The intended use of these programs results in a) generation of files in the user's UNIX directory that contain the integer-converted data, and b) generation of a 9-track digital achive tape containing the digitzed data, that can be reloaded into UNIX in the future if needed.

(1) Eaton, J.P., Playback station #2 for Calnet and 5-day-recorder tapes, U.S. Geological Survey Open File Report # 78-542, 1978. See this reference for a detailed description of the playback system.

(2) Data General Eclipse S-200 with 12-bit converter.

Brief Description of the Programs

Program FDCON (see Figure 1) controls the digitizing of five-day tapes on the Eclipse. It is modeled after the three-pass digitizing programs EVCON and EVCONPR. With the use of this program, the user specifies the data to be digitized, assigns trace names and prescribes the sampling interval and duration. The actual digitization is done by program FDDIG, which is automatically swapped in by FDCON. Program FDDIG is modeled after Program EVDIG and uses the same set of A/D-rountines.

The use of these two programs results in the creation of (or addition to) two files on disk device DP0: a directory file named EVDIR and a contiguous data file named EVDAT. These files are analogous to files produced by the three-pass EVCON program. After digitizing some data, the user of program FDCON can call another program (ISDS) which will make graphic plots of the digitized trace data. This plotting program is rather flexible but its only output device is the Tektronix 4014 display graphics terminal and hard copy unit. Some interactive picking facilities are available in ISDS.

Data file EVDAT and its associated directory file EVDIR comprise a complex data base of digitally multiplexed data that is difficult to access. Program FDDUMP is provided for selectively accessing this data base and demultiplexing the data into individual trace segments. Program FDDUMP allows a user to choose one or more events from the data base and write them, demultiplexed and in their entirety, to a 9-track digital tape. This archive tape can be appended by repeated uses of program FDDUMP.

The archive tape can be read on the PDP-11/70 UNIX system by program FDREAD, which will either load selected events onto UNIX files, or just read the tape and make a summary list of its contents.

The final file structure in the user's UNIX directory consists of 9 uniquely named files for each event. For example, the digitized data from an event starting at 79 03 27 1430 23.2 and recorded by 5-day-recorder number 66 will be contained in the following nine files:

903271430236 61
903271430236 62
903271430236 63
903271430236 64
903271430236 65
903271430236 66
903271430236 67
903271430236 68
903271430236 6h

The first eight files contain integer values of the sampled and digitized data for traces 1-8. The header file, ending in "h", contains a description of the event and the trace names. These are standard UNIX files, and can be read by programs run on UNIX, including Geolab.

Can you digitize your tape with this system?

The following discussion of the specifications and assumptions in this system, while not complete, should help answer the above question in most cases.

1. Allowable analog tapes - The software described in this report operates on either Eclipse System A or System B. The analog tape playback electronics connected to these two minicomputers differs significantly. For this reason, each playback system is treated seperately.

System A accomodates 1-inch analog tapes with 14 tracks of direct-recorder multiplexed data. Playback speed is limited by the available discriminators to 15 ips, only. Therefore this set of programs on System A can digitize Calnet and similar tapes at 15 ips, (which is 16 time recording speed).

System B accomodates 1/2-inch analog tapes (such as the five-day tapes) with six tracks of FM-recorded data and direct recorded time-code. Playback speed is limited by the available discriminators and playback filters to 15/16 ips or 3 3/4 ips. For 5-day tapes, this corresponds to 5 and 20 times recording speed, respectively).

2. A/D Sampling rate - For eight A/D channels, the maximum obtainable sampling rate is expressed below.

Playback configuration	Tape speed factor(1)	Max. sampling rate(2)
System A		
1-day tape played at 15 ips	16	240 sps
System B		
5-day tape played at 15/16 ips	5	768
5-day tape played at 3 3/4 ips	20	192

3. Number of A/D channels - The five-day tapes currently carry 8 recorded analog signals, either seismic traces or time code. The software described in this report, therefore, operates only 8 A/D channels. Attention was paid to the possibilty that more than 8 channels might be desirable in the future. Accordingly, source code was written, where possible, with the number of channels and associated variables specified as parameters. In principle these rountines could be modified to operate up to 32 A/D channels. But the amount of reprogramming time that would be required to accomplish this is not easily estimated.

(1) Tape speed factor = playback divided by recording speed.

(2) Theoretical maximum sampling rate, when this program is run under operating system USMALL. Rate is expressed in the time frame of the recorded data. The actual hardware sampling rate equals this rates times the tape speed factor.

Steps toward digitizing five-day tapes

1. Mount 1/2 -inch analog tape, and set up the playback electronics.
2. Install your disk and bootstrap operating system USMALL on the Eclipse
3. Start program FDCON.
4. (Optional) Have program FDCON list summaries of previously digitized events stored on disk.
5. Interactively enter or change the trace name assignments.
6. Digitize an event (specify event name start time, duration, sampling interval, sampling rate, etc)
7. (Optional) Make graphic plots of digitized trace data by calling plot program.
8. Repeat Steps 4-7 as necessary.
9. Exit from FDCON; start program FDDUMP (on the Eclipse).
10. Mount a 9-track digital archive tape.
11. Dump selected events from disk to the archive tape.
12. Exit from FDDUMP.
13. Mount the archive tape on PDP-11/70 and start program FDREAD on UNIX.
14. Load selected events into files in your UNIX directory.
15. Access the five-day data with the editor, Geolab, or another program on UNIX.

Structure of the FDDUMP output files on UNIX

Each event digitized on the Eclipse, if it is loaded by FDDUMP onto UNIX, results in the creation, in your UNIX directory, of nine files. The files are named:

YMMDDHHMMSSRR1
YMMDDHHMMSSRR2
YMMDDHHMMSSRR3
YMMDDHHMMSSRR4
YMMDDHHMMSSRR5
YMMDDHHMMSSRR6
YMMDDHHMMSSRR7
YMMDDHHMMSSRR8
YMMDDHHMMSSRRh

where Y = last digit of year
MM = month
DD = day
HH = hour
MM = minute
SS = second
RR = tape recorder number

and where the referenced date and time are the start of the digitization.

The first eight files contain integer-converted data samples for the eight A/D channels. Values range between -512 and 512. Each unit equals 2 of the 2.5 volt imput range, or 4.88 millivolts. These files are created with the Fortran 77 statement

```
write (9, '(16i5)') (data (k), k = 1, npts)
```

The header file (ending with "h"), contains the event unique ID, event name, date and time of first sample, duration of digitized record, sampling rate, and trace names. It is created by the following Fortran 77 code:

```
149 c
150 c
151 c        open for writing the header file and write header
152  130     open (9, file=ofile(ntr+1),form='formatted',status='new')
153 c
154          write (9,101) uid,name,imo,iday1,iyr,ihr,imin,sec,dur,
155 &                      ifix(1./delt+0.5)
156          write (9,132) (k,tname(k),k=1,ntr)
157  132     format(4(5x,i1,' = ',a4))
158          close (9)
159 c
```

```
189  101     format(/,'event id ',a16,' - ',a40,/,
190 &        'date: ',i2,'/',i2,'/',i2,14x,

191 &        'time: ',i2,':',i2,':',f5.2,/,
192 &        4x,'duration: ',f6.1,' seconds ',i3,' samples/second'\
```

Table 1

Disk file EVDIR: Structure of the 160-word fixed-length records numbered 2,3,..., N. (Record 1 is different.)

Word position	Variable type	Content
1	I	Event number (equals record # minus 1)
2-21	20I	Event name - ASCII string
22	I	Starting relative block number
23	I	Size in blocks
24-25	R	Delta-T digitizing interval, seconds
26-27	R	Duration of digitized interval, seconds
28	I	YEAR
29	I	MONTH
30	I	DAY
31	I	HOUR
32	I	MIN
33-34	R	SEC
35	I	Pointer to "Pass A paramenters" = 38
36	I	" " B " = 79
37	I	" " C " = 120
38	I	Event number
39	I	Pass number = 1
40	I	Starting relative block for this pass
41	I	Size of pass in blocks
42	I	FCA (first channel accessed by A/D)
43	I	LCA (last " " " ")
44-51	I	Pointers for 8 channels of 5-day tape
52-67	2I	4-character names of traces in this pass
68-75		Unused
76	I	Pass switch setting- actual value in digital out
77	I	Filter switch setting- actual value in digital out
78		A/D Error word (IAD/IDSK).
79-119		Pass B parameters (unused)
120-160		Pass C parameters (unused)

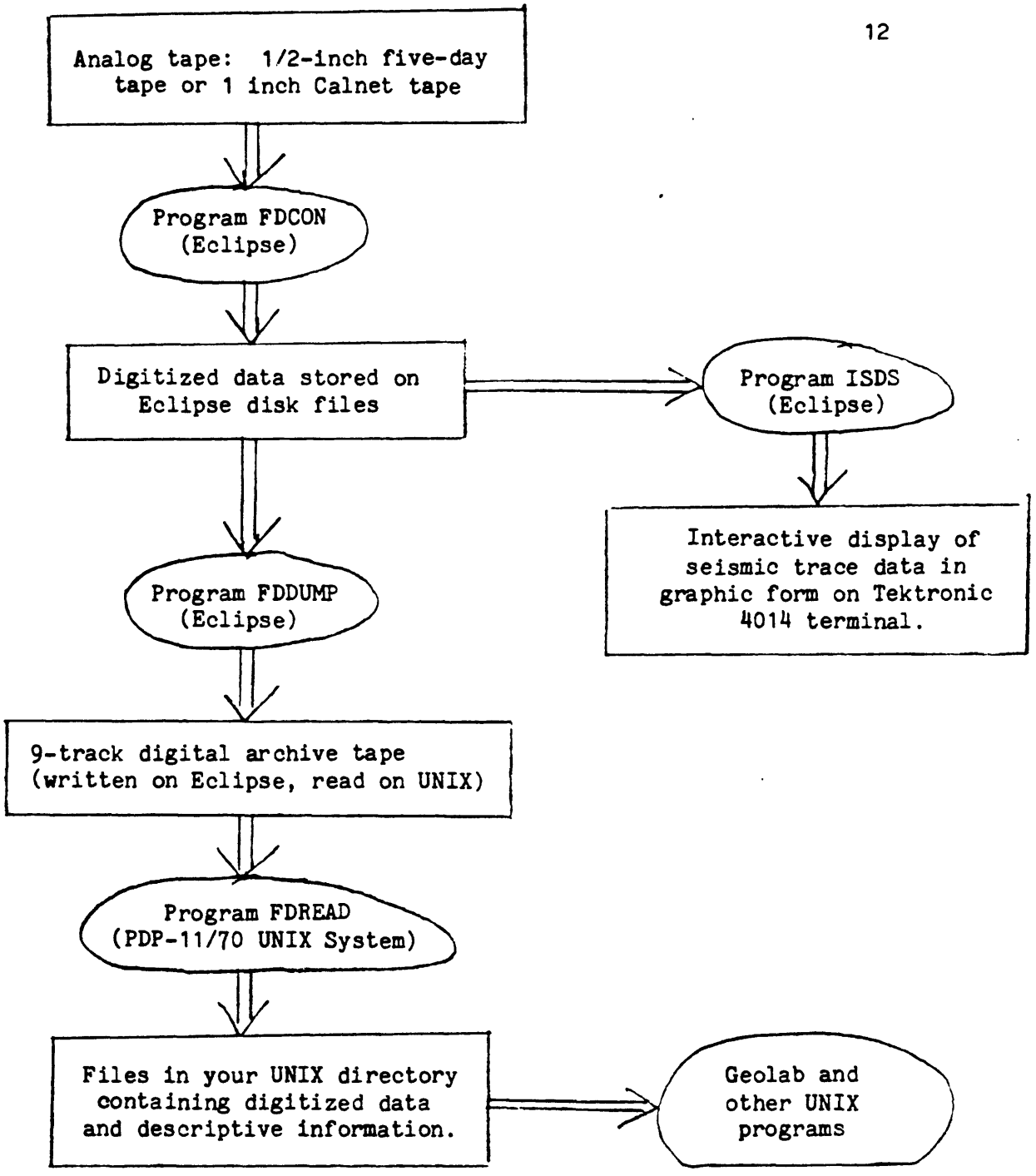


Figure 1 - Block diagram of the data paths involved in digitizing five-day tapes with the Eclipse and UNIX systems.

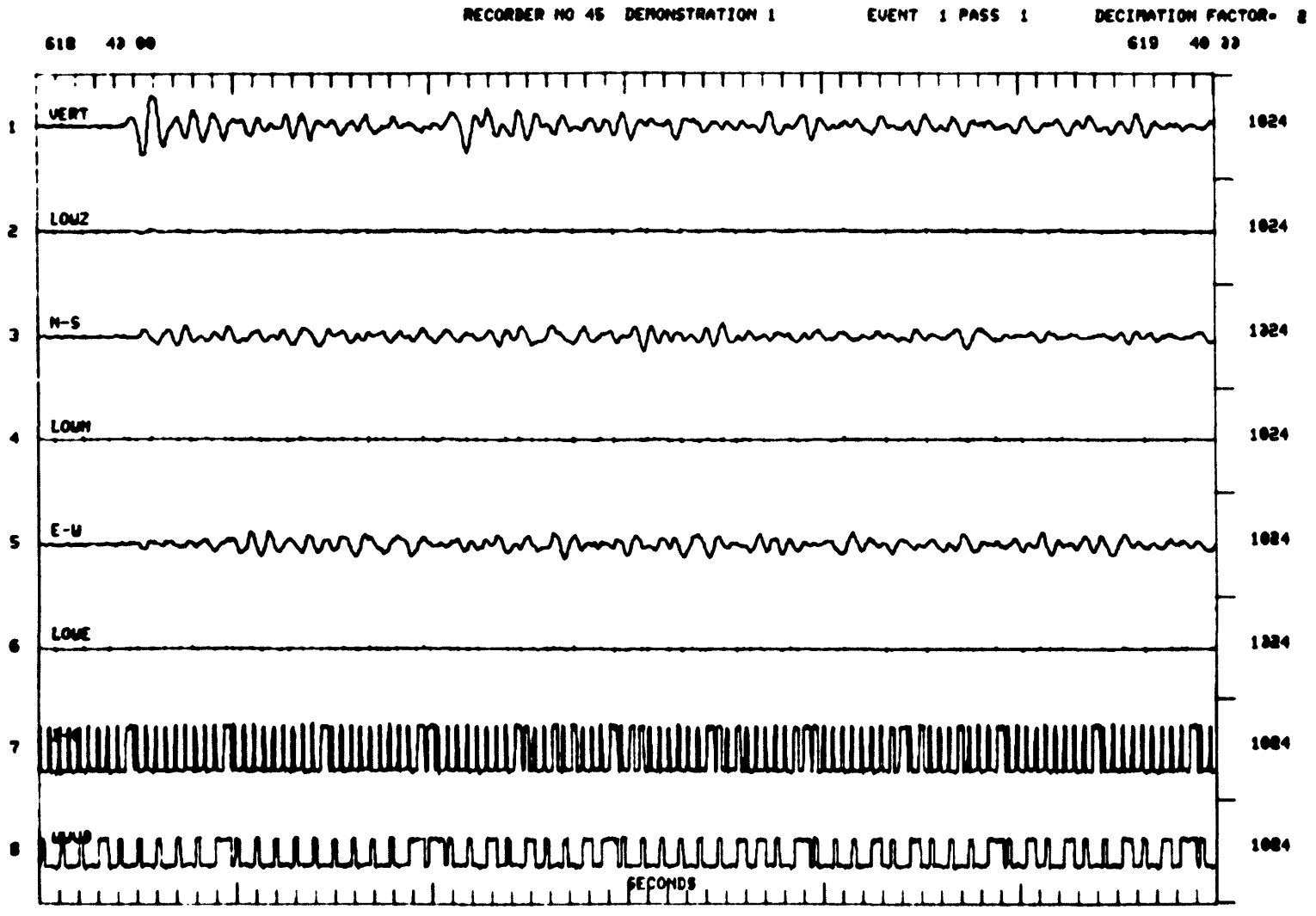
The following pages show examples of the interactive dialog required to digitize a five-day tape and bring the data through to UNIX.

```
COMMANDS

    ME - DISPLAY THIS LIST OF COMMANDS
    L  - DISPLAY A LOG OF EVENTS ON THE DISK
    MA - CHECK OR CHANGE 5-DAY TRACE NAME ASSIGNMENTS
    E  - DIGITIZE A NEW EVENT
    PL - PLOT EVENTS (USING 'ISDS')
    SP - PRINT DISK SPACE
    PI - AUTOMATIC PICKER
    OV - OVERWRITE PREVIOUSLY DIGITIZED EVENTS
    EX - EXIT FROM PROGRAM

COMMAND?
MA

LIST CURRENT TRACE NAME ASSIGNMENTS? (1=YES, 0=NO) 1

                    ---  TRACE NAME ASSIGNMENTS  ---

        1 = VERT          2 = LOWZ          3 = N-S           4 = LOWN

        5 = E-W           6 = LOWE          7 = I-C           8 = WWVB


ENTER NEW TRACE NAME ASSIGNMENTS? (1=YES, 0=NO)  1

     A/D CHANNEL    NAME

            1 ---->VERT
            2 ---->LOWZ
            3 ---->N-S
            4 ---->LOWN
            5 ---->E-W
            6 ---->LOWE
            7 ---->I-C
            8 ---->WWVB

                    ---  TRACE NAME ASSIGNMENTS  ---

        1 = VERT          2 = LOWZ          3 = N-S           4 = LOWN

        5 = E-W           6 = LOWE          7 = I-C           8 = WWVB


ENTER NEW TRACE NAME ASSIGNMENTS? (1=YES, 0=NO) 0

COMMAND?
```

Changing the trace name assignments with program FDCON. In this case, the new names and old names happen to be the same.

```
***  RECORD OF 5 DAY TAPE DIGITIZATION  ***

ENTER NAME OF EVENT (UP TO 24 CHARACTERS):
DEMONSTRATION 1
ENTER LAST TWO DIGITS OF 5-DAY RECORDER SERIAL NUMBER: 45

ENTER THE ORIGINAL RECORDING TYPE:
   ENTER 10 FOR 10-DAY-TAPES (15/180 IPS)
   ENTER  5 FOR 5-DAY-TAPES  (15/80  IPS)
   ENTER  1 FOR 1-DAY TAPES  (15/16  IPS)
--->5

ENTER PLAYBACK SPEED
   ENTER  1 FOR 15/16 IPS
   ENTER  4 FOR 3 3/4 IPS
   ENTER 16 FOR  15   IPS
   ENTER 64 FOR  60   IPS, AND SO ON
--->4


ENTER DATE OF EVENT  (YR MO DAY)  79 7 4
ENTER STARTING TIME (HR MIN SEC): 6 18 40

ENTER SAMPLES PER SECOND  150
ACTUAL A/D RATE =      150 944          DELTA T =      6 62500E-03
PLAYBACK SPEED IS       20 TIMES RECORDING SPEED
ENTER NUMBER OF SECONDS TO BE DIGITIZED: 60

                      ---  TRACE NAME ASSIGNMENTS  ---

         1 = VERT          2 = LOUZ          3 = N-S           4 = LOUN

         5 = E-W           6 = LOUE          7 = I-C           8 = UUUB


ENTER  0  TO RE-DEFINE THIS EVENT
ENTER  1  TO DIGITIZE  THIS EVENT
ENTER  2  TO RE-ASSIGN TRACE NAMES AND REDEFINE THIS EVENT
ENTER  3  TO RE-ASSIGN TRACE NAMES AND DIGITIZE THIS EVENT
--->1

EVENT       1 BEING DIGITIZED

MAKE A HARD COPY OF THIS PAGE NOW, IF NEEDED.
```

Interactive set-up for digitizing an event with program FDCON.



Plot of event digitized by previous page's dialog. Plot made by program ISDS.

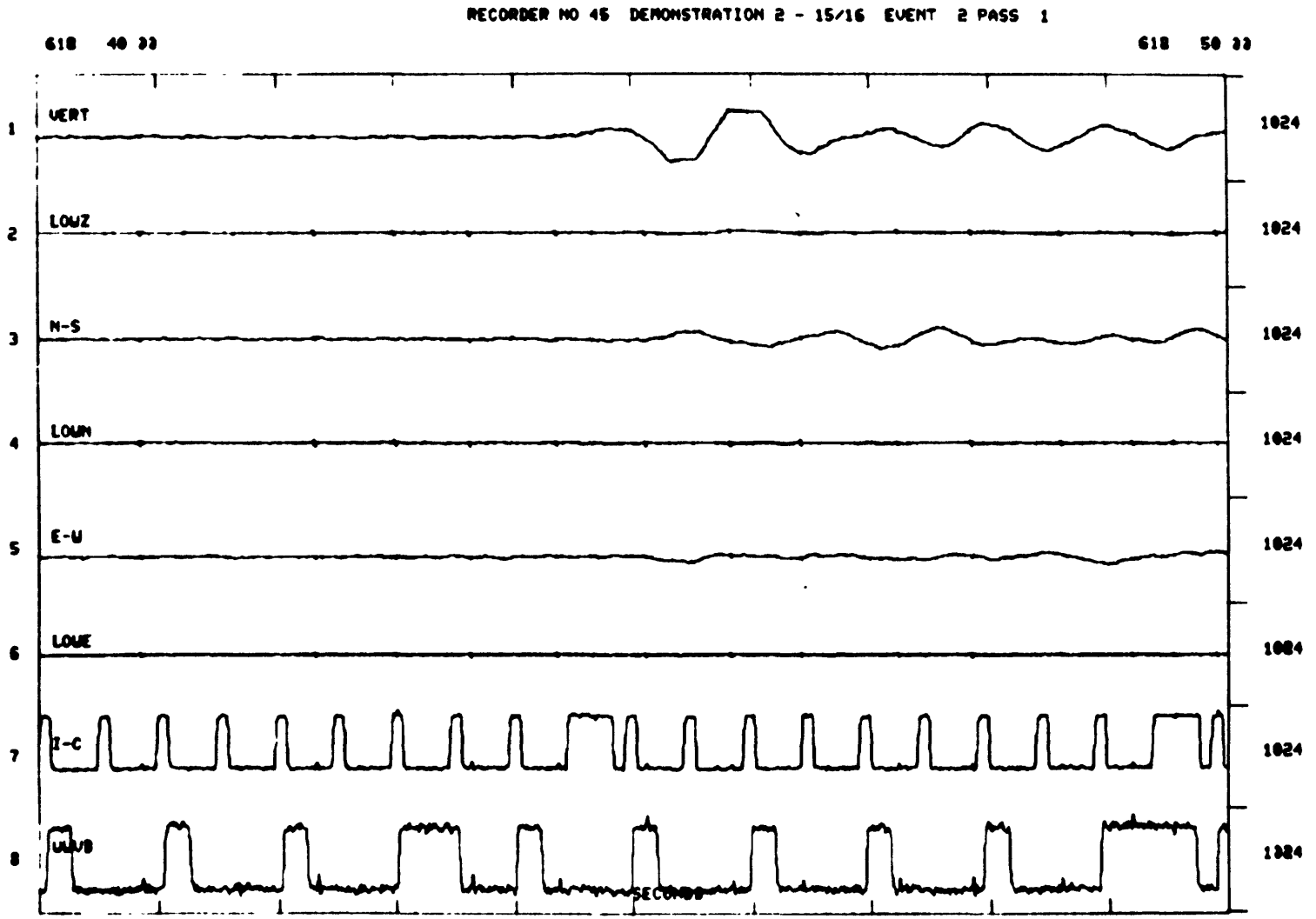
```
COMMANDS

    HE - DISPLAY THIS LIST OF COMMANDS
    L  - DISPLAY A LOG OF EVENTS ON THE DISK
    MA - CHECK OR CHANGE 5-DAY TRACE NAME ASSIGNMENTS
    E  - DIGITIZE A NEW EVENT
    PL - PLOT EVENTS (USING 'ISDS')
    SP - PRINT DISK SPACE
    PI - AUTOMATIC PICKER
    OV - OVERWRITE PREVIOUSLY DIGITIZED EVENTS
    EX - EXIT FROM PROGRAM

COMMAND?
L

      1 EVENTS ON THIS DISK

LOG WHICH EVENTS? (ENTER FIRST AND LAST): 1 1
```

Requesting a "log", or summary of the event just digitized. See next page for the result of this dialog.

```
EVENT   1        RECORDER NO 46  DEMONSTRATION 1
   DATE   7/ 4/79                 TIME  6 18 40 00        RECORDED AT 15/ 00 IPS
   DURATION  00 0 SECONDS         151 SAMPLES PER SECOND    PLAYED BACK AT X 20

                        ---  TRACE NAME ASSIGNMENTS  ---

          1 • VERT          2 • LONZ          3 • N-S           4 • LOUN

          5 • E-W           6 • LONE          7 • I-C           8 • UUUB


COMMAND?
```

```
### RECORD OF 5 DAY TAPE DIGITIZATION ###

ENTER NAME OF EVENT (UP TO 24 CHARACTERS)
[DEMONSTRATION 2 - 15/16 IPS
SELECTED CHOICE '

ENTER LAST TWO DIGITS OF 5-DAY RECORDER SERIAL NUMBER: 45

ENTER THE ORIGINAL RECORDING TYPE
   ENTER 10 FOR 10-DAY-TAPES (15/180 IPS)
   ENTER  5 FOR 5-DAY-TAPES  (15/80  IPS)
   ENTER  1 FOR 1-DAY TAPES  (15/16  IPS)
--->5

ENTER PLAYBACK SPEED
   ENTER  1 FOR 15/16 IPS
   ENTER  4 FOR 3 3/4 IPS
   ENTER 16 FOR  15   IPS
   ENTER 64 FOR  60   IPS. AND SO ON
--->1


ENTER DATE OF EVENT  (YR MO DAY): 79 7 4
ENTER STARTING TIME (HR MIN SEC)  6 18 40

ENTER SAMPLES PER SECOND  600
ACTUAL A/D RATE =      603 773          DELTA T =      1 65625E-03
PLAYBACK SPEED IS       5 TIMES RECORDING SPEED
ENTER NUMBER OF SECONDS TO BE DIGITIZED  10

                      ---  TRACE NAME ASSIGNMENTS  ---

         1 = VERT          2 = LOWZ          3 = N-S           4 = LOWN

         5 = E-W           6 = LOWE          7 = I-C           8 = WWVB


ENTER  0  TO RE-DEFINE THIS EVENT
ENTER  1  TO DIGITIZE  THIS EVENT
ENTER  2  TO RE-ASSIGN TRACE NAMES AND REDEFINE THIS EVENT
ENTER  3  TO RE-ASSIGN TRACE NAMES AND DIGITIZE THIS EVENT
--->1

EVENT       2 BEING DIGITIZED

MAKE A HARD COPY OF THIS PAGE NOW, IF NEEDED
```

Interactive dialog for digitizing the second event. This event will be digitized at 600 samples per second. Program FDCON.



Plot of the second event digitized. Program ISDS.

```
EVENT   1       RECORDER NO 45  DEMONSTRATION 1
    DATE   7/ 4/79              TIME   6 18 40 00       RECORDED AT 15/ 80 IPS
    DURATION   60 0 SECONDS     151 SAMPLES PER SECOND   PLAYED BACK AT X 20

                        ---  TRACE NAME ASSIGNMENTS  ---

          1 • VERT          2 • LONZ          3 • N-S           4 • LONN

          5 • E-W           6 • LONE          7 • I-C           8 • UUUB


EVENT   2       RECORDER NO 45  DEMONSTRATION 2 - 15/16
    DATE   7/ 4/79              TIME   6 18 40 00       RECORDED AT 15/ 80 IPS
    DURATION   10 0 SECONDS     604 SAMPLES PER SECOND   PLAYED BACK AT X  5

                        ---  TRACE NAME ASSIGNMENTS  ---

          1 • VERT          2 • LONZ          3 • N-S           4 • LONN

          5 • E-W           6 • LONE          7 • I-C           8 • UUUB


COMMAND?
```

Log, or summary, of the two events now residing on the disk files. Program FDCON.

```
COMMANDS

    ME - DISPLAY THIS LIST OF COMMANDS
    L  - DISPLAY A LOG OF EVENTS ON THE DISK
    NA - CHECK OR CHANGE 5-DAY TRACE NAME ASSIGNMENTS
    E  - DIGITIZE A NEW EVENT
    PL - PLOT EVENTS (USING 'ISDS')
    SP - PRINT DISK SPACE
    PI - AUTOMATIC PICKER
    OV - OVERWRITE PREVIOUSLY DIGITIZED EVENTS
    EX - EXIT FROM PROGRAM

COMMAND?
;P

  90 4%  DISK SPACE LEFT
COMMAND?
EX
```

Program FDCON: checking to see how much space remains in the disk file for more data. 90.4% remains, after two events.

```
WELCOME TO PROGRAM FDDUMP '      4/ 8/80     15 34

THIS PROGRAM WILL DUMP MULTIPLEXED FIVE-DAY TAPE DATA
FROM DISK FILE DP0 EVDAT TO MAGNETIC TAPE IN
DEMULTIPLEXED (TRACE BY TRACE) FORMAT

YOU CAN DUMP ALL EVENTS ON THE DISK FILE (BATCH MODE)
OR JUST SOME OF THEM (INTERACTIVE MODE)

YOU MAY CREATE EITHER BINARY RECORDS (4096 WORDS),
FOR USE ON UNIX.
OR ASCII FORMATTED (160015) RECORDS

MOUNT A WRITE-ENABLED TAPE ON TAPE UNIT ZERO.
AND ANOTHER ON TAPE UNIT ONE IF YOU EXPECT TO
FILL UP THE FIRST TAPE

 THEN STRIKE ANY KEY

THERE ARE   3 EVENTS IN THE DATABASE ON THIS DISK

PROGRAM MODE (1=BATCH, 2=INTERACTIVE)  2
TAPE FORMAT  (1=BINARY (FOR UNIX), 2=ASCII)  1


EVENT   1       RECORDER NO 45  DEMONSTRATION 1
DATE   7/ 4/79                  TIME   6 18 40 00
    DURATION    60 0 SECONDS      151 SAMPLES PER SECOND
     UNIQUE ID  79070406184045        288 DISK BLOCKS
--> STRIKE (CR) TO DUMP THIS EVENT, (SPACE) TO SKIP   <--

### WRITING TO TAPE FILE MT0 09 ###


EVENT   2       RECORDER NO 45  DEMONSTRATION 2 - 15/16
DATE   7/ 4/79                  TIME   6 18 40 00
    DURATION    10 0 SECONDS      604 SAMPLES PER SECOND
     UNIQUE ID  79070406184045        192 DISK BLOCKS
--> STRIKE (CR) TO DUMP THIS EVENT, (SPACE) TO SKIP   <--


EVENT   3       RECORDER NO 47  LV TAYLORRM4 75 12 12
DATE   3/15/80                  TIME   12 11 40 00
    DURATION    90 0 SECONDS      100 SAMPLES PER SECOND
     UNIQUE ID  80031512114047        288 DISK BLOCKS
--> STRIKE (CR) TO DUMP THIS EVENT, (SPACE) TO SKIP

NO MORE EVENTS ON DISK FILE

STOP
R
```

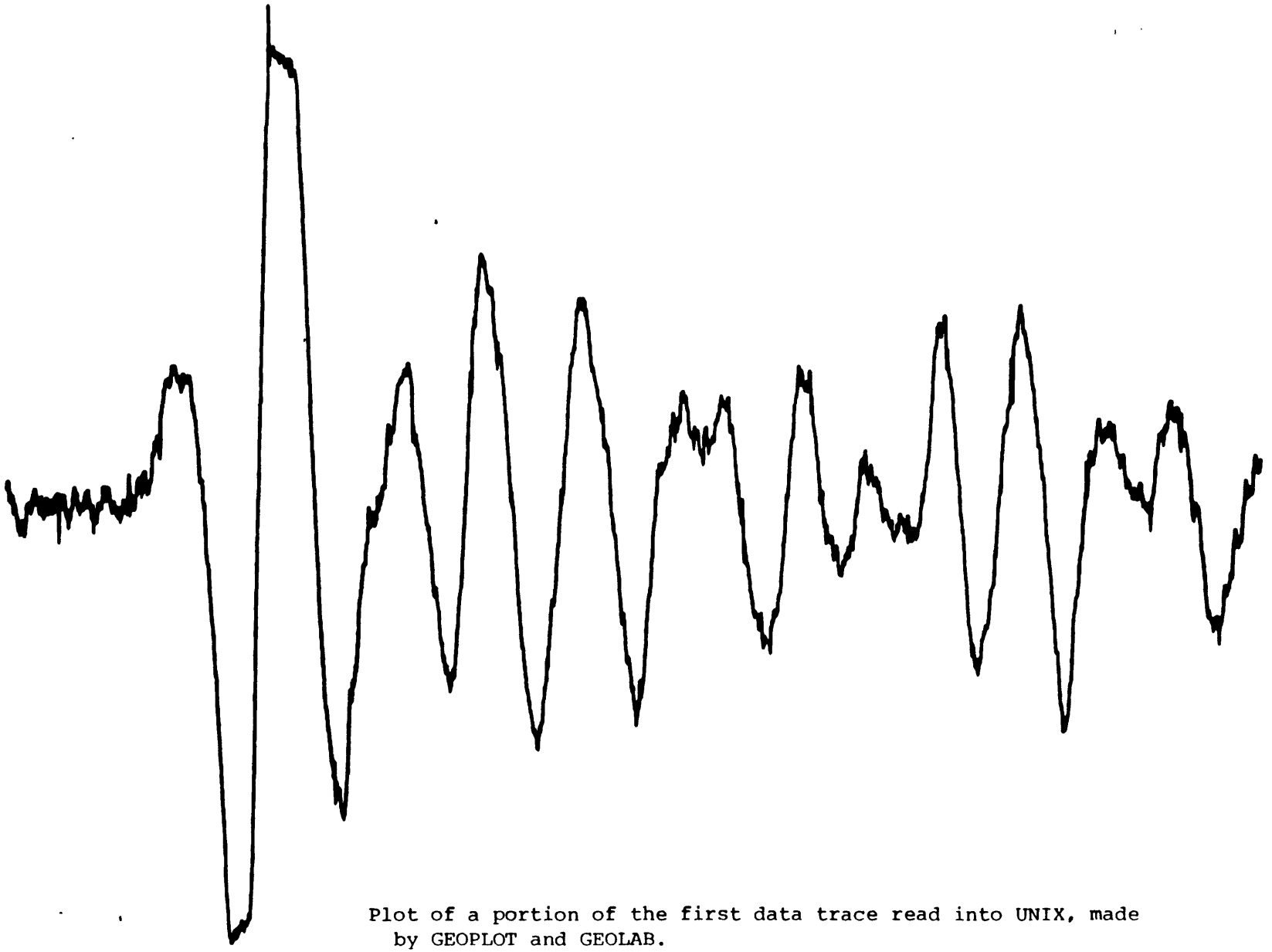
Interactive dialog from program FDDUMP. Binary tape files were requested. The first two events on the disk file were selected for transfer to tape, as indicated by the arrow on the right of each of these events. The third event was skipped. Because a previously used archive tape was used, the dumped events went to tape files 09 and 10.

```
/we/reasen/fiveday
90704061840451 90704061840455 9070406184045h geolab.w        trace1
90704061840452 90704061840456 core           geoplt
90704061840453 90704061840457 fdread         junk
90704061840454 90704061840458 fdread.out     seg1
*> cat $h

event id    7907040618404 5 - RECORDER NO.45  DEMONSTRATION 1
date:  7/ 4/79               time:  6:18: 7.23
    duration:    60.0 seconds 151 samples/second
     1 = VERT       2 = LOWZ       3 = N-S       4 = LOWN
     5 = E-W        6 = LOWE       7 = I-C       8 = WWVB
*>
*> cat $451                                 0    4    4    4   -7   -1   -6    7
  -14   -6   -8  -14  -14  -10   -4   -3    6    2   -1   -6   -5   -8   -2   -4
    5   12    6    4    7   11   12    2    0    4   -4   -1    1    2    3   -2
   -3  -10   -5   -7    8    5   12   13    3   -7   -2   -4   -5   -8    2   -2
   -7   -2    1   12   12    8   11    6   -3   -6   -1  -10   -6  -15  -18  -16
   -6   -4   -1   -2    3    1    0  -13   -4   -1   -2    5    4    0   -4    0
  -16   -8  -10   -2   -1    1   -3   -7   -3   -6    1   -4   -9  -11   -7   -6
   -8    0   -1    5    5   -1    2   -1    9    4   12   12    8   18    2  -11
   -7   -5   -2   -2   -2   -4   10    4    0    8    7    7    6    3    2    2
   13    7   16   15    3    6    0    5    8    5   -1    5   -3    2   -2    6
   13   12   12   19   21   14   20   10    5   -1    3   12    1    9    1    6
    5    6    5   -4    4   -1    4   -1    3    3    8    4   18   13   10    7
    2    2   -1   -2   -3   -3  -11   -9    3    0   -3    5   17   12   13    3
   -2   -5   -4    8    7    0    0    0   -3   -4   -6   -6   -6   -9   -5   -4
    0   -4    4    4    1   -9   -8   -7   -2    8    9    7   11    4    7    1
    9    7    4    6    2    5   -3    6    3   -3   -6   -4   -4    0    2    4
    8   10    7    6    1    8   -4    0   -4   17    5    7   10    0    1    0
    3    0    3   -1    4    5   10   12   16  -12   -7   -7    2   10   13    9
    5    0    2   -4   -7  -10   -3   -8   -2   -1    0    4    3    7    4    4
    6    2   -1   -8   -1    0    1   -6   -4   -6  -12  -11   -8  -10   -7  -15
    1    0   -4    4    3    3    0    1   -9    0    2    3    2    7   -2    2
  -10  -14   -5   -5    2   -5    1   -5    6    3    9   26   -6    2   -4    3
   -2    7    0    2    4   -2   -1   -5    6    1    5    2    2    5   -1    0
    2    9    4    0    2   -1    7    3    4    2    4    1    3   -7  -11   -2
    1    6    3    4   -4    1   -2    4    1   -6   -8   -3  -11  -14   -5   -7
   -3   -6   -8   -7   -9   -6   -6   -8  -10   -2  -10   -4  -10   -7  -12   -7
   -8  -14  -15  -18  -12  -13  -15  -11  -11    3   -1    4   -5   -4   -9  -11
   -9   -5   -8   -2   -5    3    4    5   -2   -4   -6    7   12    5    3    5
  -13   -3   -1   -4  -10   -7   -8   -6   -9   -1  -10   -6   -3   -8    0   12
    9   13    9  -11   -7   -5   -6   -4   -7   -1    3   -7   -1   -4   -4  -10
    9   11   14    8    6    1    5    3    5  -21  -20  -22  -20  -11  -13   -5
   -9  -18  -14  -19  -21  -12  -14  -15  -11    3    6    4    9   -1    2    0
   -1   -2   -6    0   -3   -6   -7   -1   -2  -11   -7   -8   -4   -2    3   -3
    3    1    2   -7   -2   -3    4   -8   -7   -4    0   -6    4    1    5    6
   -4   -8   -5   -4   -4   -3    2   -7   -7    5    4   -2   -9   -4  -11  -13
    2  -26   -7   -2    3    5   -4   -3   -2   10    0    4    6    6    1   -2
  -17  -13   -5  -11   -4   -3   -4    3    0   -4    6    0    3   -3    8    2
   -6   -4   -8   -2   -1   -4   -1   -5   -2  -14  -11   -8  -13   -8   -2    0
    2   -3   -1   -8  -11   -8   -2  -20  -11   -1   -4   -7   -5   -2    1   -2   -2   -7   -8   -5  -10   -3   -8   -4
   -2   -4   -2   -4    0    2   10         3
  -13   -7  -10   -3
*>
```

Examining the header file and one of the data files in UNIX.

```
pwd
/uo/reason/fiveday
=> ls
9070406184045l 9070406184045h 9070406184045h core          geoplt
9070406184045l 9070406184045h 9070406184045h fdread
9070406184045l 9070406184045h 9070406184045h fdread.out
=> ed 8451
46656
30,150 w segment
9801
q
=> gl
(new) Geolab -- 12 Apr 90
 ?ploton tek
Geoplot -- 15 Apr 1990
 ?open 'segment'
 ?getnums * real data!star
 ?scale count (len data) data 1 plt
```

GEOLAB and GEOPLOT: plotting a portion of the first data trace. Plot that resulted is shown on the next page.



Plot of a portion of the first data trace read into UNIX, made by GEOPLOT and GEOLAB.

APPENDIX

Program Listings

```
    1:
    2:      C       PROGRAM FDCOM - FIVE-DAY-RECORDER TAPE DIGITIZING PROGRAM
    3:      C
    4:      C                       TO RUN ON THE USGS ECLIPSE SYSTEMS UNDER
    5:      C                       OPERATING SYSTEM 'USMALL'
    6:      C
    7:      C
    8:      C       THIS PROGRAM IS A MODIFICATION OF THE THREE-PASS DIGITIZING
    9:      C       PROGRAM 'EVCON' AND 'EVCOMPR' TO ACCOMODATE THE FIVE-DAY
   10:      C       TAPE REQUIREMENTS.
   11:      C
   12:      C       THE ORIGINAL DIGITIZING PROGRAMS WERE DEVELOPED BY P STEVENSON,
   13:      C       P.A.MARSHALL, T.THOMAS, AND T.JACKSON IN 1977
   14:      C       AND HAVE BEEN MODIFIED SEVERAL TIMES SINCE THEN. THE
   15:      C       A TO D ROUTINES USED ARE STILL THE ORIGINAL ROUTINES WRITTEN
   16:      C       BY MARSHALL AND THOMAS.
   17:      C
   18:      C
   19:      C       PAUL REASENBERG, APRIL, 1980.
   20:      C
   21:      C*********************************************************************
   22:      C
   23:      C----
   24:              INCLUDE 'EVPARM.AD'
   25:      C*********************************************************************
   26:      C  PARAMETERS AND COMMON FOR THE EVCON PACKAGE
   27:      C
   28:      C
   29:      C
   30:      C***** GENERAL PARAMETERS *****
   31:              INTEGER DRLN,DRLNB,SWITCHLUN,FILTLUN
   32:              LOGICAL NODUB
   33:              PARAMETER DRLN=160, DRLNB=DRLN*2
   34:              PARAMETER MLLN=40, MLLNB=2*MLLN
   35:              PARAMETER NUMPASSES=3,SWITCHLUN=6,FILTLUN=5
   36:      C  DRLN=LENGTH OF A RECORD IN THE EVENT DIRECTORY (EVDIR.AD)
   37:      C    DRLNB=THE SAME RECORD LENGTH AS DRLN, BUT IN BYTES
   38:      C  MLLN=LENGTH OF A RECORD IN THE MASTERLIST DIRECTORY (MASTERLIST.AD)
   39:      C    MLLNB= THE SAVE RECORD LENGTH AS MLLN, BUT IN BYTES
   40:      C    THE RECORD LENGTH IN BYTES IS USED WHEN OPENING THESE FILES
   41:      C  NUMPASSES=NUMBER OF PASSES NEEDED TO DIGITIZE ALL OF THE TAPE TRACKS
   42:      C
   43:      C       (EXPLAIN SET UP OF PASSES)
   44:      C
   45:      C  SWITCHLUN=LOGICAL UNIT NUMBER (LUN) OF THE OUTPUT BITS THAT CONTROL
   46:      C     THE TRACK SELECT SWITCH
   47:      C  FILTLUN=L U.N  OF THE OUTPUT BITS THAT CONTROL THE FILTER SELECT SWITCH
   48:      C
   49:      C       (EXPLAIN SWITCH SELECTION)
   50:      C
   51:      C
   52:      C
   53:      C
   54:      C***** DEFINE COMMON AREA *****
   55:              INTEGER STATUS,EVNT
   56:              COMMON /A2D/ STATUS(DRLN),EVNT(DRLN),LIN(80)
```

```
57:   C
58:   C
59:   C
60:   C***** DEFINE STATUS ARRAY LAYOUT *****
61:           INTEGER PNTR(96),TIM2AD,FILSIZ
62:           EQUIVALENCE (NEV,STATUS(1)),(NMLST,STATUS(2)),(IOFF,STATUS(3)),
63:        *    (FILSIZ,STATUS(4)),(NXTBLK,STATUS(5)),(TIM2AD,STATUS(6)),(IDELT,STATUS(7)),
64:        *    (IMEX,STATUS(8)),(NODUB,STATUS(9)),(SCANS,STATUS(10)),
65:        *    (PNTR(1),STATUS(12))
66:   C  THE FIRST RECORD IN THE DIRECTORY FILE (DPO:EVDIR) CONTAINS
67:   C     GENERAL INFORMATION AND IS ALWAYS KEPT IN ARRAY "STATUS".
68:   C     THE LAYOUT OF THIS RECORD IS AS FOLLOWS:
69:   C ARRAY
70:   C INDEX NAME    DESCRIPTION
71:   C   1   NEV     NUMBER OF EVENTS ALREADY ON DISK
72:   C   2   NMLST   NUMBER OF ENTRIES IN MASTERLIST FILE
73:   C   3   IOFF    INITIAL OFFSET FOR CONTIGUOUS FILE "DPO:EVDAT"
74:   C   4   FILSIZ  TOTAL USABLE SPACE IN "DPO:EVDAT", IN DISK BLOCKS
75:   C   5   NXTBLK  NUMBER OF THE NEXT DISK BLOCK IN "DPO:EVDAT" TO START
76:   C                 STORE DATA INTO (INITIALLY EQUALS "IOFF")
77:   C   6   TIM2AD  THE NUMBER OF THE A/D CHANNEL RECIEVING A TIME CODE
78:   C                 0 IF NO TIME CODE
79:   C   7   IDELT   TIME PARAMETER FOR P.I.T. THERE ARE IDELT*6.25
80:   C                 MICROSECONDS BETWEEN SAMPLE POINTS
81:   C   8   IMEX    FOOTAGE COUNT FOR TAPE POSITIONING FOR THE CURRENT EVENT
82:   C   9   NODUB   TRUE IF THE TAPE BEING DIGITIZED IS NOT A DUBBED TAPE
83:   C  10   SCANS   THE (REAL) NUMBER OF SAMPLES TO BE DIGITIZED ON THE
84:   C                 CURRENT EVENT
85:   C  12   PNTR    AN ARRAY USED TO MAP TAPE TRACK-CHANNEL TO A RECORD
86:   C                 IN THE MASTERLIST FILE, WHICH IS SUPPOSED TO BE A
87:   C                 STATION CARD IMAGE
88:   C
89:   C
90:   C
91:   C***** DEFINE EVNT ARRAY LAYOUT *****
92:           INTEGER FCA,GPSIZ,PPSIZ,ADERR,TSSWV,FSSWV,SIZPS,SIZEV,EVNUM,
93:        *    EVNUMPS,PSNUM,PSPNT,TIM(7),NAM(20),IPASS(NUMPASSES)
94:           PARAMETER EVNUMPS=0,PSNUM=1,IBLKPS=2,SIZPS=3,FCA=4,LCA=5
95:           PARAMETER PSPNT=6,TSSWV=38,FSSWV=39,ADERR=40,PPSIZ=ADERR+1
96:           EQUIVALENCE (EVNUM,EVNT(1)),(NAM(1),EVNT(2)),(IBLKEV,EVNT(22)),
97:        *    (SIZEV,EVNT(23)),(DELT,EVNT(24)),(DUR,EVNT(26)),
98:        *    (TIM(1),EVNT(28)),(SEC,EVNT(33)),(IPASS(1),EVNT(35))
99:           PARAMETER GPSIZ=38
100:  C
101:  C  (DISCRIBE SETUP)
102:  C
103:  C
104:  C*************************************************************************
105:  C
106:          DIMENSION ICM(4)
107:  C         ICM WILL HOLD THE COMMAND STRING OF 2 CHARACTERS AND A NULL BYTE
108:  C
109:    100   OPEN 10,"$TT01"
110:          OPEN 11,"$TTI1"
111:          OPEN 5,"$TTI1"
112:          OPEN 2, "DPO:MASTERLIST", LEN=MLLNG
113:          CALL INIT("DPO",IER)
114:          IF ((IER.NE.1).AND.(IER.NE.40)) CALL BOMB(1,IER)
115:  C         OPEN TERMINAL I/O, AND TRY TO INITIALIZE DPO  IF DPO CAN'T BE
116:  C           INITIALIZED AND ISN'T ALREADY THERE, BOMB OUT!
```

```
C
      CALL INITT(0)
      CALL SETBUF(4)
      CALL CHRSIZ(3)
      CALL FDELY(15)
      TYPE "** WELCOME TO THE ECLIPSE 5-DAY TAPE DIGITIZING PROGRAM **"
C
      CALL STAT("DP0:EVDAT",EVNT,IER)
      IF (IER.NE.1) GOTO 200
C       TRY TO GET THE 18 WORD STATUS ARRAY FOR EVDAT, IF EVDAT EXISTS
C
      IF (ITEST(EVNT(7),3).EQ.0) CALL BOMB(3,IER)
C       MAKE SURE IT'S A CONTIGUOUS FILE
C
      OPEN 1,"DP0:EVDIR",LEN=DRLNB,ERR=250
             READ(1,REC=1) STATUS
             I=MOD(EVNT(11),12)
             IF (I.NE.0) I=12-I
             IF (I.NE.IOFF) GOTO 255
             IF (FILSIZ.NE.(((EVNT(9)+1-IOFF)/12)*12)) GOTO 255
C               MAKE SURE THE FILE PARAMETERS MATCH UP!
             IF (NEV.LE.1) GOTO 260
 460        CONTINUE
             TYPE " "
             TYPE NEV-1," EVENTS ALREADY ON DISK"
             TYPE " "
             CALL FDELY(5)
             CALL DISKSPACE
             CALL FDELY(5)
             TYPE " "
             ACCEPT "DISPLAY LOG OF EVENTS? (1=YES, 0=NO): ",I
             IF (I.EQ.1) CALL FDLOG
 110         ACCEPT "ENTER NUMBER OF FIRST EVENT FOR THIS SESSION: ",NEXT
             IF ((NEXT.GT.NEV).OR.(NEXT.LT.1)) GOTO 110
             NDEAD=NEV-NEXT
             IF(NDEAD.LE.0)GOTO 471
             TYPE "DELETE LAST ",NDEAD," EVENTS? (1=YES, 0=NO)"
             TYPE " "
             READ FREE(11,END=10)I
             IF(I.NE.1)GOTO 110
 471        CONTINUE
C              NEV=NEXT
C              IF (NEV.LE.1) GOTO 270
C              READ(1,REC=NEV) EVNT
C              NXTBLK=IBLKEV+SIZEV
C                RESET THE STARTING BLOCK NUMBER
C              CALL DISKSPACE
C              GOTO 300
C
C
C
C
 200    CONTINUE
C
C  IF EVDAT DOESN'T EXIST, TRY TO CREATE IT (CONTIGUOUS, OF COURSE!)
C  SINCE THERE MUST BE ROOM FOR THE DIRECTORY FILE, DON'T MAKE IT ANY
C  BIGGER THAN 9000 DISK BLOCKS, BUT IT SHOULD BE BIGGER THAN 200
C  BLOCKS TO BE OF ANY USE.  OTHER THAN THAT, MAKE IT AS BIG AS POSSIBLE!
      DO 210 I=9000,200,-6
             CALL CFILW("DP0:EVDAT",3,I,IER)
```

```
177:                  IF (IER.EQ.1) GOTO 220
178:  C                 IF FILE CREATED SUCCESSFULLY, GET OUT OF LOOP
179:    210        CONTINUE
180:               CALL BOMB(2,IER)
181:               STOP
182:  C               EVDAT NEVER CREATED SUCCESSFULLY SO BOMB OUT
183:  C
184:    220    CALL STAT("DPO:EVDAT",EVNT,IER)
185:  C            EVDAT CREATED SUCCESSFULLY, SO GET STATUS ARRAY
186:  C
187:  C
188:  C
189:  C
190:    250      CONTINUE
191:           OPEN 1,"DPO:EVDIR",LEN=DRLNB
192:    255    NEV=1
193:           NMLST=0
194:           IOFF=MOD(EVNT(11),12)
195:           IF (IOFF.NE.0) IOFF=12-IOFF
196:           FILSIZ=((EVNT(9)+1-IOFF)/12)*12
197:    260    TYPE "THERE ARE NO EVENTS STORED ON THIS DISK"
198:    270    NXTBLK=IOFF
199:           CALL DISKSPACE
200:  C
201:  C
202:  C        ASSIGN THE TRACE NAMES FOR THE 5-DAY TAPE
203:           CALL FDNLPL
204:           NODUB = .TRUE.
205:  C
206:  C
207:  C**********************************************
208:  C*********** COMMAND LOOP *********************
209:  C**********************************************
210:  C
211:    300      CONTINUE
212:           TYPE "COMMANDS:"
213:           TYPE " "
214:           TYPE "     ME - DISPLAY THIS LIST OF COMMANDS"
215:           TYPE "     L  - DISPLAY A LOG OF EVENTS ON THE DISK"
216:           TYPE "     NA - CHECK OR CHANGE 5-DAY TRACE NAME ASSIGNMENTS"
217:           TYPE "     E  - DIGITIZE A NEW EVENT"
218:           TYPE "     PL - PLOT EVENTS (USING 'ISDS')"
219:           TYPE "     SP - PRINT DISK SPACE"
220:           TYPE "     PI - AUTOMATIC PICKER"
221:           TYPE "     OV - OVERWRITE PREVIOUSLY DIGITIZED EVENTS"
222:           TYPE "     EX - EXIT FROM PROGRAM"
223:           TYPE " "
224:  C
225:    310      CONTINUE
226:           WRITE(1,REC=1) STATUS
227:           TYPE " "
228:    311    TYPE "COMMAND?"
229:    320    READ(11,3,END=900) ICM(1)
230:      3    FORMAT(S2)
231:           TYPE " "
232:           IF (ICM(1).EQ."EX") GOTO 900
233:           IF (ICM(1).EQ."NA") GOTO 400
234:           IF ((ICM(1).AND.177400K).EQ."L(000)") GOTO 410
235:           IF ((ICM(1).AND.177400K).EQ."E(000)") GOTO 420
236:           IF (ICM(1).EQ."PL") GOTO 430
```

```
237:            IF(ICM(1).EQ."SP")GOTO 450
238:            IF(ICM(1).EQ."PI")GOTO 430
239:            IF (ICM(1).EQ."ME") GOTO 440
240:            IF(ICM(1).EQ."OV")GOTO 460
241:            TYPE "HUH? "
242:            GOTO 320
243:    C
244:      400   CALL FDMLPL
245:            GOTO 310
246:    C
247:      410   CALL FDLOG
248:            GOTO 310
249:    C
250:      420   CALL FDEVENT
251:            GOTO 310
252:    C
253:      430     CONTINUE
254:            CALL UPDATE(1,IER)
255:            CALL SINGLETASK
256:            IF(ICM(1).EQ."PI")CALL FSWAP("PICKER.SV")
257:            IF(ICM(1).EQ."PL")CALL FSWAP("ISDS.SV")
258:            CALL MULTITASK
259:            CALL MTRST
260:            CALL RECOVR
261:            CALL NEWPAG
262:            READ (1,REC=1) STATUS
263:            TYPE "** 5-DAY TAPE DIGITIZING PROGRAM -- FDCON **"
264:            TYPE " "
265:            GOTO 310
266:    C
267:      450   CONTINUE
268:            CALL FDELY(5)
269:            CALL DISKSPACE
270:            GOTO 311
271:      440   CALL NEWPAG
272:            CALL FDELY(10)
273:            GOTO 300
274:    C
275:    C
276:    C
277:       10     CONTINUE
278:            CALL RESET
279:            GOTO 100
280:    C
281:    C
282:      900   CALL RESET
283:            STOP
284:            END
```

	ATTRIBUTES	POSITION	SIZE
-- A2D --			
STATUS	INTEGER ARRAY	0	160.
PNTR	INTEGER ARRAY	13	96.
SCANS	REAL	11	
MODUB	LOGICAL	10	
IHEX	INTEGER	7	
IDELT	INTEGER	6	
TIM2AD	INTEGER	5	
NXTBLK	INTEGER	4	
FILSIZ	INTEGER	3	
IOFF	INTEGER	2	
NNLST	INTEGER	1	
NEV	INTEGER	0	
EVNT	INTEGER ARRAY	240	160.
IPASS	INTEGER ARRAY	302	3.
SEC	REAL	300	
TIM	INTEGER ARRAY	273	7.
DUR	REAL	271	
DELT	REAL	267	
SIZEV	INTEGER	266	
IBLKEV	INTEGER	265	
NAM	INTEGER ARRAY	241	20.
EVNUM	INTEGER	240	
LIN	INTEGER ARRAY	300	80.
-- STACK VARIABLES --			
I	INTEGER	1	
NEXT	INTEGER	2	
NDEAD	INTEGER	3	
IER	INTEGER	4	
ICM	INTEGER ARRAY	5	4.

-- EXTERNAL SUBPROGRAMS --

.FOPE	.ILEN	.MCAL	INIT	BOMB
INITT	SETBU	CHRSI	FDELY	ITYP
.FWPS	.TTYP	STAT	ITEST	IERR
.SEEK	.IBRD	.BRDI	.TBRD	.FWRI
DISKS	.IACC	.FRDI	.TACC	FDLOG
.IEMD	.IURD	TURD	CFILW	STOP
FDMLP	.IBWR	.BWRI	.TBWR	IFFD
.TFRD	FDEVE	UPDAT	SINGL	FSWAP
MULTI	HTRST	RECOV	NEWFA	RESET

IDENTIFIER	REFERENCES
ADERR	95 93
BONB	114 128 180
CFILW	176
CHRSIZ	120
DELT	98
DISKSPACE	144 163 199 269
DRLN	33 31
DRLNB	33 31
DUR	98
EVNT	55 56 98 124 128 133 136 160 184 194 196
EVNUM	93 98
EVNUMPS	94 93
FCA	94 93
FDELY	121 143 145 268 272
FDEVENT	250
FDLOG	148 247
FDMLPL	203 244
FILSIZ	61 65 136 196
FILTLUN	35 31
FSSUV	95 93
FSWAP	256 257
GPSIZ	99 93
HTRST	259
I	133 134 135 147 148 155 156 175 176 179
IBLKEV	98 161
IBLKPS	94
ICN	106 229 232 233 234 235 236 237 238 239 240 256 257
IDELT	65
IER	113 114 124 125 128 176 177 180 184 254
INEX	65
INIT	113
INITT	118
IOFF	65 135 136 194 195 196 198
IPASS	93 98
ITEST	128
LCA	94
LIN	56
MLLN	34
MLLNB	34
MULTITASK	258
NAM	93 98
NDEAD	151 152 153
NEV	65 138 141 150 151 158 159 160 192
NEWPAG	261 271
NEXT	149 150 151 158
NMLST	65 193
NODUB	32 65 204
NUMPASSES	35
NXTBLK	65 161 198
PNTR	61 65
PPSIZ	95 93
PSNUM	94 93
PSPNT	95 93
RECOVR	260
RESET	278 282
SCANS	65
SEC	98

```
NO COMPILATION ERRORS  --  TERMINATED AT 3:27:58 PM
```

FDNAMES.FR

```
      SUBROUTINE FDNAMES(NUMBER,IP,IPNT,I,L,IPATH)
C
C       WRITTEN BY      P.REASENBERG
C                       OCTOBER, 1979
C
C       SUBROUTINE TO WRITE THE STATION NAMES FOR THE
C       5-DAY TAPE DIGITIZATION PROGRAM TO THE TEKTRONIX 4014
C       TERMINAL.  THIS SUBROUTINE IS CALLED BY SUBROUTINE 5DVENT,
C       WHICH IS CALLED, IN TURN, BY PROGRAM 5DCON.
C
C       THIS ROUTINE IS A MODIFICAATION OF A SIMILAR ROUTINE CALLED
C       SUBROUTINE NAMES, WRITTEN FOR THE SAME PURPOSE FOR
C       THE "3-PASS" ECLIPSE DIGITIZING PROGRAM, EVCONPR.
C
C*************************************************************************
C
      INCLUDE 'EVPARM.AD'
C*************************************************************************
C  PARAMETERS AND COMMON FOR THE EVCON PACKAGE
C
C
C
C***** GENERAL PARAMETERS *****
      INTEGER DRLN,DRLNB,SWITCHLUN,FILTLUN
      LOGICAL MODUB
      PARAMETER DRLN=160, DRLNB=DRLN*2
      PARAMETER MLLN=40, MLLNB=2*MLLN
      PARAMETER NUMPASSES=3,SWITCHLUN=6,FILTLUN=5
C  DRLN=LENGTH OF A RECORD IN THE EVENT DIRECTORY (EVDIR.AD)
C    DRLNB=THE SAME RECORD LENGTH AS DRLN, BUT IN BYTES
C  MLLN=LENGTH OF A RECORD IN THE MASTERLIST DIRECTORY (MASTERLIST.AD)
C    MLLNB= THE SAVE RECORD LENGTH AS MLLN, BUT IN BYTES
C    THE RECORD LENGTH IN BYTES IS USED WHEN OPENING THESE FILES
C  NUMPASSES=NUMBER OF PASSES NEEDED TO DIGITIZE ALL OF THE TAPE TRACKS
C
C       <EXPLAIN SET UP OF PASSES>
C
C  SWITCHLUN=LOGICAL UNIT NUMBER (LUN) OF THE OUTPUT BITS THAT CONTROL
C     THE TRACK SELECT SWITCH
C  FILTLUN=L.U.N. OF THE OUTPUT BITS THAT CONTROL THE FILTER SELECT SWITCH
C
C       <EXPLAIN SWITCH SELECTION>
C
C
C
C
C***** DEFINE COMMON AREA *****
      INTEGER STATUS,EVNT
      COMMON /A2D/ STATUS(DRLN),EVNT(DRLN),LIN(80)
C
C
C
C***** DEFINE STATUS ARRAY LAYOUT *****
      INTEGER PNTR(96),TIM2AD,FILSIZ
      EQUIVALENCE (NEV,STATUS(1)),(NMLST,STATUS(2)),(IOFF,STATUS(3)),
     *    (FILSIZ,STATUS(4)),(NXTBLK,STATUS(5)),(TIM2AD,STATUS(6)),(IDELT,STATUS(7)),
```

```
57:        *    (INEX,STATUS(8)),(NODUB,STATUS(9)),(SCANS,STATUS(10)),
58:        *    (PNTR(1),STATUS(12))
59:     C  THE FIRST RECORD IN THE DIRECTORY FILE (DPO:EVDIR) CONTAINS
60:     C     GENERAL INFORMATION AND IS ALWAYS KEPT IN ARRAY "STATUS"
61:     C     THE LAYOUT OF THIS RECORD IS AS FOLLOWS:
62:     C ARRAY
63:     C INDEX NAME     DESCRIPTION
64:     C   1   NEV      NUMBER OF EVENTS ALREADY ON DISK
65:     C   2   NMLST    NUMBER OF ENTRIES IN MASTERLIST FILE
66:     C   3   IOFF     INITIAL OFFSET FOR CONTIGUOUS FILE "DPO:EVDAT"
67:     C   4   FILSIZ   TOTAL USABLE SPACE IN "DPO:EVDAT", IN DISK BLOCKS
68:     C   5   NXTBLK   NUMBER OF THE NEXT DISK BLOCK IN "DPO:EVDAT" TO START
69:     C                  STORE DATA INTO (INITIALLY EQUALS "IOFF")
70:     C   6   TIM2AD   THE NUMBER OF THE A/D CHANNEL RECIEVING A TIME CODE.
71:     C                  0 IF NO TIME CODE
72:     C   7   IDELT    TIME PARAMETER FOR P.I.T. THERE ARE IDELT*6.25
73:     C                  MICROSECONDS BETWEEN SAMPLE POINTS
74:     C   8   INEX     FOOTAGE COUNT FOR TAPE POSITIONING FOR THE CURRENT EVENT
75:     C   9   NODUB    TRUE IF THE TAPE BEING DIGITIZED IS NOT A DUBBED TAPE
76:     C  10   SCANS    THE (REAL) NUMBER OF SAMPLES TO BE DIGITIZED ON THE
77:     C                  CURRENT EVENT
78:     C  12   PNTR     AN ARRAY USED TO MAP TAPE TRACK-CHANNEL TO A RECORD
79:     C                  IN THE MASTERLIST FILE, WHICH IS SUPPOSED TO BE A
80:     C                  STATION CARD IMAGE
81:     C
82:     C
83:     C
84:     C***** DEFINE EVNT ARRAY LAYOUT *****
85:           INTEGER FCA,GPSIZ,PPSIZ,ADERR,TSSWV,FSSWV,SIZPS,SIZEV,EVNUM,
86:          *   EVNUMPS,PSNUM,PSPNT,TIM(7),NAM(20),IPASS(NUMPASSES)
87:           PARAMETER EVNUMPS=0,PSNUM=1,IBLKPS=2,SIZPS=3,FCA=4,LCA=5
88:           PARAMETER PSPNT=6,TSSWV=38,FSSWV=39,ADERR=40,PPSIZ=ADERR+1
89:           EQUIVALENCE (EVNUM,EVNT(1)),(NAM(1),EVNT(2)),(IBLKEV,EVNT(22)),
90:          *   (SIZEV,EVNT(23)),(DELT,EVNT(24)),(DUR,EVNT(26)),
91:          *   (TIM(1),EVNT(28)),(SEC,EVNT(33)),(IPASS(1),EVNT(35))
92:           PARAMETER GPSIZ=38
93:     C
94:     C  (DISCRIBE SETUP)
95:     C
96:     C
97:     C*************************************************************************
98:     C
99:           EQUIVALENCE (NAME(1),EVNT(52))
100:          DIMENSION IPNT(32),IML(MLLN),NAME(16)
101:    C
102:    C
103:         8   WRITE (10,1)
104:         1   FORMAT (/24X,"---  TRACE NAME ASSIGNMENTS  ---"//,8X,Z)
105:    C
106:          II=1
107:          LL=8
108:          DO 100 LN=1,L,LL
109:          DO 110 N=LN,MIN0(LN+LL-1,L)
110:          IF (IPATH.EQ.1) GOTO 30
111:    C
112:    C
113:    C     GET TRACE NAMES FROM DPO:MASTERLIST (PATH=2)
114:          IF (IPNT(N).LT.1) GOTO 10
115:          READ(2,REC=IPNT(N)) IML
116:          ENCODE(LIM,2) II,IML(1),IML(2)
```

```
117:            GOTO 35
118:    C
119:    C       GET TRACE NAMES FROM ARRAY EVNT (PATH=1)
120:       30   ENCODE (LIN,2) II,NAME(II*2-1),NAME(II*2)
121:       35   WRITE (10,3) (LIN(K),K=1,9)
122:        3   FORMAT (9A2,Z)
123:        2   FORMAT(1X,I2," = ",2S2,8X)
124:            GOTO 20
125:       10   CONTINUE
126:            ENCODE(LIN,4) II
127:            WRITE (10,3) (LIN(K),K=1,9)
128:        4   FORMAT(1X,I2," = ....",8X)
129:       20   CONTINUE
130:            II=II+1
131:            IF (II.NE.5) GOTO 110
132:            WRITE (10,5)
133:        5   FORMAT (//8X,Z)
134:      110   CONTINUE
135:            WRITE (10,5)
136:      100   CONTINUE
137:            WRITE (10,6)
138:        6   FORMAT (/)
139:            GOTO 9000
140:      990   TYPE *FDNAMES: END OF FILE ON 1*
141:            GOTO 9000
142:      991   TYPE *FDNAMES: ERROR DURING READ --  REC =*,REC
143:     9000   RETURN
144:            END
145:
```

	ATTRIBUTES	POSITION	SIZE
-- DUMMY ARGUMENTS --			
NUMBER	SUBROUTINE	0-3	
IP	SUBROUTINE	0-4	
IPNT	INTEGER ARRAY	0-5	
I	INTEGER	0-6	
L	INTEGER	0-7	
IPATH	INTEGER	0-10	
-- A2D --			
STATUS	INTEGER ARRAY	0	160.
PNTR	INTEGER ARRAY	13	96.
SCANS	REAL	11	
NODUB	LOGICAL	10	
INEX	INTEGER	7	
IDELT	INTEGER	6	
TIM2AD	INTEGER	5	
NXTBLK	INTEGER	4	
FILSIZ	INTEGER	3	
IOFF	INTEGER	2	
NNLST	INTEGER	1	
NEV	INTEGER	0	
EVNT	INTEGER ARRAY	240	160.
NAME	INTEGER ARRAY	323	16.
IPASS	INTEGER ARRAY	302	3.
SEC	REAL	300	
TIM	INTEGER ARRAY	273	7.
DUR	REAL	271	
DELT	REAL	267	
SIZEV	INTEGER	266	
IBLKEV	INTEGER	265	
NAM	INTEGER ARRAY	241	20.
EVNUM	INTEGER	240	
LIN	INTEGER ARRAY	300	80.
-- STACK VARIABLES --			
LN	INTEGER	1	
LL	INTEGER	2	
M	INTEGER	3	
II	INTEGER	4	
K	INTEGER	5	
INL	INTEGER ARRAY	6	40.
REC	REAL	56	

-- EXTERNAL SUBPROGRAMS --

.IFWR	.TFWR	.SEEK	.IBRD	.BRDI
.TBRD	.IENC	.FWRI	.TENC	MINO
.ITYP	.FWRS	.TTYP	.FWRR	

IDENTIFIER	REFERENCES					
ADERR	88	86				
DELT	91					
DRLN	26	24				
DRLNB	26	24				
DUR	91					
EVNT	48	49	91	99		
EVNUM	86	91				
EVNUMPS	87	86				
FCA	87	86				
FDNAMES	1					
FILSIZ	54	58				
FILTLUN	20	24				
FSSUV	88	86				
GPSIZ	92	86				
I	1	108				
IBLKEV	91					
IBLKPS	87					
IDELT	58					
IHEX	58					
II	106	116	120	126	130	131
IML	100	115	116			
IOFF	58					
IP	1					
IPASS	86	91				
IPATH	1	110				
IPNT	1	100	114	115		
K	121	127				
L	1	108	109			
LCA	87					
LIN	49	116	120	121	126	127
LL	107	108	109	136		
LN	108	109	136			
MIND	109					
MLLN	27					
MLLNB	27					
N	109	114	115	134		
NAM	86	91				
NAME	100	99	120			
NEV	58					
NNLST	58					
NODUB	25	58				
NUMBER	1					
NUMPASSES	28					
NXTBLK	58					
PNTR	54	58				
PPSIZ	88	86				
PSNUM	87	86				
PSPNT	88	86				
REC	142					
SCANS	58					
SEC	91					
SIZEV	86	91				
SIZPS	87	86				
STATUS	48	49	58			
SWITCHLUN	28	24				
TIM	86	91				
TIM2AD	54	58				

```
TSSWV       88    86

1           104   103
10          114   125
100         108   136
110         109   131   134
2           123   116   120
20          124   129
3           122   121   127
30          110   120
35          117   121
4           128   126
5           133   132   135
6           138   137
8           103
9000        139   141   143
990         140
991         142

A2D         49
```

FDEVENT.FR


```
      SUBROUTINE FDEVENT
C
      INCLUDE 'EVPARM.AD'
C**********************************************************************
C  PARAMETERS AND COMMON FOR THE EVCON PACKAGE
C
C
C
C***** GENERAL PARAMETERS *****
      INTEGER DRLN,DRLNB,SWITCHLUN,FILTLUN
      LOGICAL NODUB
      PARAMETER DRLN=168, DRLNB=DRLN*2
      PARAMETER NLLN=40, NLLNB=2*NLLN
      PARAMETER NUMPASSES=3,SWITCHLUN=6,FILTLUN=5
C  DRLN=LENGTH OF A RECORD IN THE EVENT DIRECTORY (EVDIR.AD)
C    DRLNB=THE SAME RECORD LENGTH AS DRLN, BUT IN BYTES
C  NLLN=LENGTH OF A RECORD IN THE MASTERLIST DIRECTORY (MASTERLIST.AD)
C    NLLNB= THE SAVE RECORD LENGTH AS NLLN, BUT IN BYTES
C    THE RECORD LENGTH IN BYTES IS USED WHEN OPENING THESE FILES
C  NUMPASSES=NUMBER OF PASSES NEEDED TO DIGITIZE ALL OF THE TAPE TRACKS
C
C       <EXPLAIN SET UP OF PASSES>
C
C  SWITCHLUN=LOGICAL UNIT NUMBER (LUN) OF THE OUTPUT BITS THAT CONTROL
C     THE TRACK SELECT SWITCH
C  FILTLUN=L.U.N. OF THE OUTPUT BITS THAT CONTROL THE FILTER SELECT SWITCH
C
C       <EXPLAIN SWITCH SELECTION>
C
C
C
C
C***** DEFINE COMMON AREA *****
      INTEGER STATUS,EVNT
      COMMON /A2D/ STATUS(DRLN),EVNT(DRLN),LIN(80)
C
C
C
C***** DEFINE STATUS ARRAY LAYOUT *****
      INTEGER PNTR(96),TIM2AD,FILSIZ
      EQUIVALENCE (NEV,STATUS(1)),(NMLST,STATUS(2)),(IOFF,STATUS(3)),
     *   (FILSIZ,STATUS(4)),(NXTBLK,STATUS(5)),(TIM2AD,STATUS(6)),(IDELT,STATUS(7)),
     *   (INEX,STATUS(8)),(NODUB,STATUS(9)),(SCANS,STATUS(10)),
     *   (PNTR(1),STATUS(12))
C  THE FIRST RECORD IN THE DIRECTORY FILE (DPO:EVDIR) CONTAINS
C    GENERAL INFORMATION AND IS ALWAYS KEPT IN ARRAY "STATUS".
C    THE LAYOUT OF THIS RECORD IS AS FOLLOWS:
C ARRAY
C INDEX NAME    DESCRIPTION
C   1   NEV     NUMBER OF EVENTS ALREADY ON DISK
C   2   NMLST   NUMBER OF ENTRIES IN MASTERLIST FILE
C   3   IOFF    INITIAL OFFSET FOR CONTIGUOUS FILE "DPO:EVDAT"
C   4   FILSIZ  TOTAL USABLE SPACE IN "DPO:EVDAT", IN DISK BLOCKS
C   5   NXTBLK  NUMBER OF THE NEXT DISK BLOCK IN "DPO:EVDAT" TO START
C                 STORE DATA INTO (INITIALLY EQUALS "IOFF")
C   6   TIM2AD  THE NUMBER OF THE A/D CHANNEL RECIEVING A TIME CODE.
```

```
97:   C                  0 IF NO TIME CODE
98:   C    7   IDELT   TIME PARAMETER FOR P.I.T. THERE ARE IDELT*6.25
99:   C                  MICROSECONDS BETWEEN SAMPLE POINTS
100:  C    8   IHEX    FOOTAGE COUNT FOR TAPE POSITIONING FOR THE CURRENT EVENT
101:  C    9   NODUB   TRUE IF THE TAPE BEING DIGITIZED IS NOT A DUBBED TAPE
102:  C   10   SCANS   THE (REAL) NUMBER OF SAMPLES TO BE DIGITIZED ON THE
103:  C                  CURRENT EVENT
104:  C   12   PNTR    AN ARRAY USED TO MAP TAPE TRACK-CHANNEL TO A RECORD
105:  C                  IN THE MASTERLIST FILE, WHICH IS SUPPOSED TO BE A
106:  C                  STATION CARD IMAGE
107:  C
108:  C
109:  C
110:  C***** DEFINE EVNT ARRAY LAYOUT *****
111:        INTEGER FCA,GPSIZ,PPSIZ,ADERR,TSBUV,FSBUV,SIZPS,SIZEV,EVNUM,
112:       *   EVNUMPS,PSNUM,PSPNT,TIM(7),NAM(20),IPASS(NUMPASSES)
113:        PARAMETER EVNUMPS=0,PSNUM=1,IBLKPS=2,SIZPS=3,FCA=4,LCA=5
114:        PARAMETER PSPNT=6,TSBUV=38,FSBUV=39,ADERR=40,PPSIZ=ADERR+1
115:        EQUIVALENCE (EVNUM,EVNT(1)),(NAM(1),EVNT(2)),(IBLKEY,EVNT(22)),
116:       *   (SIZEV,EVNT(23)),(DELT,EVNT(24)),(DUR,EVNT(26)),
117:       *   (TIM(1),EVNT(28)),(SEC,EVNT(33)),(IPASS(1),EVNT(35))
118:        PARAMETER GPSIZ=38
119:  C
120:  C  <DISCRIBE SETUP>
121:  C
122:  C
123:  C**********************************************************************
124:  C
125:        INTEGER FSW(3),PSW(3),NEX(2),TSF,BLANK/'  '/,FDAY
126:        DATA PSW/3,1,2/,FSW/3,0,1/
127:        DIMENSION IML(MLLN),NAME(16)
128:        EQUIVALENCE (NAME(1),EVNT(52))
129:        EQUIVALENCE (ISPEED,EVNT(85)),(IRS,EVNT(86))
130:  C
131:     10 WRITE (10,12)
132:     12 FORMAT ('<33><14>')
133:        CALL FDELY(15)
134:        IBLKEY=NXTBLK
135:        EVNUM=NEV
136:  C
137:        DO 200 I=1,3
138:        IPASS(I)=GPSIZ+(I-1)*PPSIZ
139:        EVNT(IPASS(I)+EVNUMPS)=NEV
140:        EVNT(IPASS(I)+PSNUM)=I
141:        EVNT(IPASS(I)+ADERR)=0
142:  C     THE FOLLOWING LINE WAS CHANGED FROM J=0,31 BECAUSE
143:  C     THE LOCATIONS CORRESPONDING TO J=8,31 ARE NOW USED FOR
144:  C     STORING THE 8 TRACE NAMES FOR 5-DAY EVENTS.
145:        DO 210 J=0,7
146:        EVNT(IPASS(I)+PSPNT+J)=PNTR((I-1)*32+J+1)
147:    210 CONTINUE
148:    200 CONTINUE
149:  C
150:  C     SET UP THE DIGITIZATION SPECIFICATIONS FOR THIS EVENT
151:     10 DO 105 I=9,20
152:    105 NAM(I)=BLANK
153:  C
154:    120 CALL FGTIME(IHR,IMIN,ISEC)
155:        TYPE '***  RECORD OF 5 DAY TAPE DIGITIZATION  ***'
156:        TYPE ' '
```

```
      TYPE "ENTER NAME OF EVENT (UP TO 24 CHARACTERS): "
      READ (11,7,END=10) NAM(9)
    7 FORMAT (924)
    8 IF (MOD(ISEC,10).NE.2) GOTO 9
      WRITE (10,122)
  122 FORMAT ('SPLENDID CHOICE !'/)
    9 ACCEPT "ENTER LAST TWO DIGITS OF 5-DAY RECORDER SERIAL NUMBER: ",ISDTP
      IF (ISDTP.LT.0.OR.ISDTP.GT.99) GOTO 9
      ENCODE (NAM,4) ISDTP
    4 FORMAT ('RECORDER NO.',I2,2X)
C
C
C     ESTABLISH THE RECORD AND PLAYBACK TAPE SPEEDS.
      WRITE (10,115)
  115 FORMAT (/"ENTER THE ORIGINAL RECORDING TYPE:"/
     &        "   ENTER 10 FOR 10-DAY-TAPES (15/180 IPS)"/
     &        "   ENTER  5 FOR 5-DAY-TAPES  (15/80  IPS)"/
     &        "   ENTER  1 FOR 1-DAY TAPES  (15/16  IPS)")
  116 ACCEPT '--->',IRS
      IF (IRS.NE.1.AND.IRS.NE.5.AND.IRS.NE.10) GOTO 116
      WRITE (10,118)
  118 FORMAT (/'ENTER PLAYBACK SPEED:'/
     &        '   ENTER  1 FOR 15/16 IPS'/
     &        '   ENTER  4 FOR 3 3/4 IPS'/
     &        '   ENTER 16 FOR  15   IPS'/
     &        '   ENTER 64 FOR  60   IPS, AND SO ON...')
  119 ACCEPT '--->',IPS
      TYPE " "
      IF (IPS.NE.1.AND.IPS.NE.2.AND.IPS.NE.4.AND.IPS.NE.8.
     &    AND.IPS.NE.16.AND.IPS.NE.32.AND.IPS.NE.64) GOTO 119
C
C     TSF IS THE 'TAPE SPEED FACTOR' WHICH IS DEFINED AS THE MULTIPLICATIVE
C     FACTOR BY WHICH RECORDED VCO CARRIERS ARE TRANSFORMED IN FREQUENCY.
      TSF = IPS*IRS
C
      ISPEED=IPS
      IF (NODUB) GOTO 100
   90 ACCEPT "ENTER HEX FOOTAGE COUNT: ",HEX
      IHEX=UNHEX(HEX)
      IF (IHEX.EQ.-1) GOTO 90
  100 CONTINUE
      TYPE " "
      ACCEPT "ENTER DATE OF EVENT  (YR MO DAY): ",TIM(1),TIM(2),TIM(3)
      ACCEPT "ENTER STARTING TIME (HR MIN SEC): ",TIM(4),TIM(5),SEC
      TYPE " "
  101 ACCEPT "ENTER SAMPLES PER SECOND: ",SPS
      IF (SPS.LT.1) GOTO 101
C
      IDELT=160000/(SPS*TSF)
      DELT=FLOAT(TSF*IDELT)/160000.
      TYPE "ACTUAL A/D RATE = ",1./DELT,"      DELTA T = ",DELT
      TYPE "PLAYBACK SPEED IS ",TSF," TIMES RECORDING SPEED"
      ACCEPT "ENTER NUMBER OF SECONDS TO BE DIGITIZED: ",DUR
C
      SCANS=AINT(DUR/DELT)+1.
C
C
C     DISPLAY TRACE NAME ASSIGNMENTS FOR PASS 1 ONLY
      I=1
      IF=1
```

```
177:          IL=0
178:          EVNT(IPASS(I)+TSSUV)=PSW(I)
179:          IMODE=2
180:          CALL FDNAMES(NEV,I,EVNT(IPASS(I)+PSPNT),IF,IL,IMODE)
181:          EVNT(IPASS(I)+FCA)=IF
182:          EVNT(IPASS(I)+LCA)=IL
183:    C     SET UP TO SAMPLE A/D CHANNELS 1 THROUGH 8 (SKIP PASSES 2 AND 3)
184:          EVNT(IPASS(2)+FCA)=0
185:          EVNT(IPASS(2)+LCA)=0
186:          EVNT(IPASS(3)+FCA)=0
187:          EVNT(IPASS(3)+LCA)=0
188:          EVNT(IPASS(I)+FSSUV)=FSW(1)
189:    C
190:          WRITE (10,103)
191:      103 FORMAT ('ENTER  0  TO RE-DEFINE THIS EVENT'/
192:         &          'ENTER  1  TO DIGITIZE  THIS EVENT'/
193:         &          'ENTER  2  TO RE-ASSIGN TRACE NAMES AND REDEFINE THIS EVENT'/
194:         &          'ENTER  3  TO RE-ASSIGN TRACE NAMES AND DIGITIZE THIS EVENT')
195:      112 ACCEPT '---)',IBRANCH
196:          IF (IBRANCH.LT.0.OR.IBRANCH.GT.3) GOTO 112
197:          GOTO (15,104,106,108) IBRANCH+1
198:      106 CALL FDNLPL
199:          GOTO 15
200:      108 CALL FDNLPL
201:          GOTO 104
202:      104 CONTINUE
203:          TYPE * *
204:          CALL FDELY(10)
205:    C
206:    C     LOAD ARRAY EVNT WITH THE CURRENT TRACE NAMES
207:          DO 126 K=1,8
208:          READ (2,REC=K) IML
209:          NAME(K*2-1) = IML(1)
210:      126 NAME(K*2)   = IML(2)
211:    C
212:    C     ENSCRIBE THESE PRECIOUS BITS OF DATA ON THE DISK
213:          TYPE 'EVENT ',NEV,' BEING DIGITIZED...'
214:          CALL FDELY(20)
215:          TYPE * *
216:          TYPE "MAKE A HARD COPY OF THIS PAGE NOW, IF NEEDED...."
217:          WRITE(1,REC=NEV+1) EVNT
218:          WRITE(1,REC=1) STATUS
219:    C
220:    C     DO THE DIGITIZING...
221:          CALL SINGLETASK
222:          CALL UPDATE(1,IER)
223:          CALL FSWAP("FDDIG.SV")
224:          CALL MULTITASK
225:          CALL NTRST
226:          CALL RECOVR
227:    C
228:    C     CHECK DPO:EVDIR FOR SIGNS OF DIGITIZING ERRORS IN PASS 1
229:          READ(1,REC=1) STATUS
230:          READ(1,REC=NEV) EVNT
231:          I=1
232:          IF (EVNT(IPASS(I)+ADERR).EQ.0) GOTO 300
233:          IAD=EVNT(IPASS(I)+ADERR)
234:          IDSK=FLD(IAD,9,16)
235:          IAD=FLD(IAD,1,0)
236:          WRITE (10,13)
```

```
237:            CALL FDELY(15)
238:            WRITE (10,11) I+100K+20000K, IAD, IDSK
239:         11 FORMAT ('DIGITIZING ERROR, PASS',I2,', IAD=',I4,', IDSK=',I4/)
240:            TYPE 'HIT (CR) WHEN YOU ARE READY'
241:            READ (11,14) JPAUSE
242:         14 FORMAT (A1)
243:        300 CONTINUE
244:      C
245:            WRITE (10,13)
246:         13 FORMAT ('<33><14>')
247:            CALL FDELY(15)
248:            CALL DISKSPACE
249:            RETURN
250:            END
251:
```

ATTRIBUTES		POSITION	SIZE
-- A2D --			
STATUS	INTEGER ARRAY	0	160.
PNTR	INTEGER ARRAY	13	96.
SCANS	REAL	11	
NODUB	LOGICAL	10	
INEX	INTEGER	7	
IDELT	INTEGER	6	
TIN2AD	INTEGER	5	
NXTBLK	INTEGER	4	
FILSIZ	INTEGER	3	
IOFF	INTEGER	2	
ANLST	INTEGER	1	
NEV	INTEGER	0	
EVNT	INTEGER ARRAY	240	160.
IRS	INTEGER	365	
ISPEED	INTEGER	364	
NAME	INTEGER ARRAY	323	16.
IPASS	INTEGER ARRAY	302	3.
SEC	REAL	300	
TIM	INTEGER ARRAY	273	7.
DUR	REAL	271	
DELT	REAL	267	
SIZEV	INTEGER	266	
IBLKEV	INTEGER	265	
NAM	INTEGER ARRAY	241	20.
EVNUM	INTEGER	240	
LIN	INTEGER ARRAY	500	00.
-- STATIC VARIABLES --			
FSW	INTEGER ARRAY	0	3.
PSW	INTEGER ARRAY	3	3.
BLANK	INTEGER	6	
-- STACK VARIABLES --			
TSF	INTEGER	1	
FDAY	INTEGER	2	
I	INTEGER	3	
J	INTEGER	4	
ISBTP	INTEGER	5	
IPS	INTEGER	6	
IBRANCH	INTEGER	7	
K	INTEGER	10	
IAD	INTEGER	11	
IDSK	INTEGER	12	
JPAUSE	INTEGER	13	
INR	INTEGER	14	
IMIN	INTEGER	15	
ISEC	INTEGER	16	
IF	INTEGER	17	
IL	INTEGER	20	
IMODE	INTEGER	21	

```
IER       INTEGER                 22
NEX       INTEGER ARRAY           23          2.
INL       INTEGER ARRAY           25          40.
SPS       REAL                    75

-- EXTERNAL SUBPROGRAMS --

.IFUR           .TFUR           .NCAL           FDELY           FCTIM
.ITYP           .FURS           .TTYP           .IEND           .IFRD
.FRDI           .TFRD           .IACC           .TACC           .IENC
.FURI           .TENC           UNNEX           .FRDR           .FURR
AINT            FDNAN           .CGO            FDNLP           .SEEK
.IBRD           .BRDI           .TBRD           .ISUR           .BURI
.TBUR           SINGL           UPDAT           FSWAP           MULTI
ATRST           RECOV           .FLDR           DISKS
```

```
IDENTIFIER                    REFERENCES

ADERR       74   72
AINT        171
BLANK       85   112
DELT        77   166  167  171
DISKSPACE   240
DRLN        12   10
DRLNB       12   10
DUR         77   169  171
EVNT        34   35   77   88   89   99   100  101  106  178  180  181  182  184
            185  186  187  188  217  230  232  233
EVNUM       72   77   95
EVNUMPS     73   72
FCA         73   72
FDAY        85
FDELY       93   204  214  237  247
FDEVENT     1
FDNLPL      190  200
FDNAMES     190
FGTIME      114
FILSIZ      40   44
FILTLUN     14   18
FSSUV       74   72
FSU         85   86   190
FSWAP       223
GPSIZ       70   72
HEX         89   154  155
HTRST       225
I           97   98   99   100  101  106  188  111  112  175  178  180  181  182
            188  231  232  233  238
ISETP       123  124  125
IAD         233  234  235  238
IDLKEV      77   94
IDLKPS      73
IDRANCH     195  196  197
IDELT       44   165  166
IDSK        234  238
IER         222
IF          176  180  181
INEX        44   155  156
INR         114
IL          177  180  182
ININ        114
INL         87   208  209  210
INODE       179  180
IOFF        44
IPASS       72   77   98   99   100  101  106  178  180  181  182  184  185  186
            187  188  232  233
IPS         143  146  150  152
IRS         89   135  136  150
ISEC        114  120
ISPEED      89   152
J           185  186  187
JPAUSE      241
K           207  208  209  210
LCA         73
LIN         35
MLLN        13
```

FDMLPL.FR

```
      SUBROUTINE FDMLPL
C
C
C       WRITTEN BY:     P. REASENBERG
C                       U.S.G.S.
C                       OCTOBER, 1979
C
C       PURPOSE:        ASSIGNS TRACE NAMES FOR 5-DAY TAPE DIGITIZING
C                       PROGRAM, FDCON.  THIS ROUTINE INTERACTIVELY
C                       ASSIGNS TRACE NAMES, WRITING THEM TO
C                       DP0:MASTERLIST, AND POINTERS.  THE ROUTINE
C                       ASSUMES ONLY ONE PASS IS BEING DIGITIZED, AND
C                       THAT ONLY THE FIRST EIGHT CHANNELS OF PASS 1
C                       ARE IN USE.
C
C
C*****************************************************************************
C
      INTEGER BLANK/'  '/
      DIMENSION NLENTD(41),NLENT(40),NAME(16)
      EQUIVALENCE (NAME(1),EVNT(32))
      EQUIVALENCE (NLENTD(1),NLENT(1))
C
      INCLUDE 'EVPARM.AD'
C*****************************************************************************
C  PARAMETERS AND COMMON FOR THE EVCOM PACKAGE
C
C
C
C***** GENERAL PARAMETERS *****
      INTEGER DRLN,DRLNB,SWITCHLUN,FILTLUN
      LOGICAL NODUB
      PARAMETER DRLN=160, DRLNB=DRLN*2
      PARAMETER NLLN=40, NLLNB=2*NLLN
      PARAMETER NUMPASSES=3,SWITCHLUN=6,FILTLUN=5
C  DRLN=LENGTH OF A RECORD IN THE EVENT DIRECTORY (EVDIR.AD)
C    DRLNB=THE SAME RECORD LENGTH AS DRLN, BUT IN BYTES
C  NLLN=LENGTH OF A RECORD IN THE MASTERLIST DIRECTORY (MASTERLIST.AD)
C    NLLNB= THE SAVE RECORD LENGTH AS NLLN, BUT IN BYTES
C    THE RECORD LENGTH IN BYTES IS USED WHEN OPENING THESE FILES
C  NUMPASSES=NUMBER OF PASSES NEEDED TO DIGITIZE ALL OF THE TAPE TRACKS
C
C       <EXPLAIN SET UP OF PASSES>
C
C  SWITCHLUN=LOGICAL UNIT NUMBER (LUN) OF THE OUTPUT BITS THAT CONTROL
C     THE TRACK SELECT SWITCH
C  FILTLUN=L.U.N. OF THE OUTPUT BITS THAT CONTROL THE FILTER SELECT SWITCH
C
C       <EXPLAIN SWITCH SELECTION>
C
C
C
C
C***** DEFINE COMMON AREA *****
      INTEGER STATUS,EVNT
      COMMON /A2D/ STATUS(DRLN),EVNT(DRLN),LIN(80)
```

```
C
C
C
C***** DEFINE STATUS ARRAY LAYOUT *****
      INTEGER PNTR(96),TIM2AD,FILSIZ
      EQUIVALENCE (NEV,STATUS(1)),(MMLST,STATUS(2)),(IOFF,STATUS(3)),
     *   (FILSIZ,STATUS(4)),(NXTBLK,STATUS(5)),(TIM2AD,STATUS(6)),(IDELT,STATUS(7)),
     *   (INEX,STATUS(8)),(NODUB,STATUS(9)),(SCANS,STATUS(10)),
     *   (PNTR(1),STATUS(12))
C  THE FIRST RECORD IN THE DIRECTORY FILE (DPO:EVDIR) CONTAINS
C     GENERAL INFORMATION AND IS ALWAYS KEPT IN ARRAY "STATUS".
C     THE LAYOUT OF THIS RECORD IS AS FOLLOWS:
C ARRAY
C INDEX NAME    DESCRIPTION
C   1   NEV     NUMBER OF EVENTS ALREADY ON DISK
C   2   MMLST   NUMBER OF ENTRIES IN MASTERLIST FILE
C   3   IOFF    INITIAL OFFSET FOR CONTIGUOUS FILE "DPO:EVDAT"
C   4   FILSIZ  TOTAL USABLE SPACE IN "DPO:EVDAT", IN DISK BLOCKS
C   5   NXTBLK  NUMBER OF THE NEXT DISK BLOCK IN "DPO:EVDAT" TO START
C                 STORE DATA INTO (INITIALLY EQUALS "IOFF")
C   6   TIM2AD  THE NUMBER OF THE A/D CHANNEL RECIEVING A TIME CODE.
C                 0 IF NO TIME CODE
C   7   IDELT   TIME PARAMETER FOR P.I.T. THERE ARE IDELT*6.25
C                 MICROSECONDS BETWEEN SAMPLE POINTS
C   8   INEX    FOOTAGE COUNT FOR TAPE POSITIONING FOR THE CURRENT EVENT
C   9   NODUB   TRUE IF THE TAPE BEING DIGITIZED IS NOT A DUBBED TAPE
C  10   SCANS   THE (REAL) NUMBER OF SAMPLES TO BE DIGITIZED ON THE
C                 CURRENT EVENT
C  12   PNTR    AN ARRAY USED TO MAP TAPE TRACK-CHANNEL TO A RECORD
C                 IN THE MASTERLIST FILE, WHICH IS SUPPOSED TO BE A
C                 STATION CARD IMAGE
C
C
C
C***** DEFINE EVNT ARRAY LAYOUT *****
      INTEGER FCA,GPSIZ,PPSIZ,ADERR,TSSUV,FSSUV,SIZPS,SIZEV,EVNUM,
     *   EVNUMPS,PSNUM,PSPNT,TIM(7),NAM(20),IPASS(NUMPASSES)
      PARAMETER EVNUMPS=0,PSNUM=1,IBLKPS=2,SIZPS=3,FCA=4,LCA=5
      PARAMETER PSPNT=6,TSSUV=38,FSSUV=39,ADERR=40,PPSIZ=ADERR+1
      EQUIVALENCE (EVNUM,EVNT(1)),(NAM(1),EVNT(2)),(IBLKEV,EVNT(22)),
     *   (SIZEV,EVNT(23)),(DELT,EVNT(24)),(DUR,EVNT(26)),
     *   (TIM(1),EVNT(29)),(SEC,EVNT(33)),(IPASS(1),EVNT(35))
      PARAMETER GPSIZ=38
C
C  <DISCRIBE SETUP>
C
C
C*********************************************************************
C
C
C
      IF (NEV.LE.1) GOTO 50
   40 ACCEPT "LIST CURRENT TRACE NAME ASSIGNMENTS? (1=YES, 0=NO): ",ILIST
      IF (ILIST.NE.1) GOTO 50
C
C     LIST - GET TRACE NAMES FROM DPO:MASTERLIST
      IMODE=2
   45 CALL FDNAMES (NEV,1,EVNT(IPASS(1)+PSPNT),1,0,IMODE)
   50 ACCEPT "ENTER NEW TRACE NAME ASSIGNMENTS? (1=YES, 0=NO): ",ICHANGE
      IF (ICHANGE.NE.1) GOTO 900
```

```
117:    C
118:    C       CHANGE - GET TRACE NAMES FROM CONSOLE OPERATOR
119:      100   DO 110 I=1,96
120:      110   PNTR(I)=0
121:            NMLST=0
122:    C
123:            WRITE (10,3)
124:        3   FORMAT (/5X,'A/D CHANNEL    NAME'/)
125:    C
126:            DO 120 I=1,8
127:            DO 121 J=1,40
128:      121   NLENT(J)=BLANK
129:      119   WRITE (10,1) I
130:        1   FORMAT (13X,I1,' ---->',Z)
131:            READ (11,2,END=119) NLENT(1)
132:        2   FORMAT (80A)
133:            NMLST=NMLST+1
134:            WRITE (2,REC=NMLST) NLENT
135:    C
136:    C       LOAD ARRAY EVNT WITH TRACE NAMES FOR CURRENT EVENT
137:            EVNT(IPASS(1)+PSPNT+I-1)=I
138:            NAME(I*2-1) = NLENT(1)
139:            NAME(I*2)   = NLENT(2)
140:            PNTR(I)=I
141:            IF (NLENT(1).EQ.BLANK.AND.NLENT(2).EQ.BLANK)
142:         &     EVNT(IPASS(1)+I-1)=0
143:            IF (NLENT(1).EQ.BLANK.AND.NLENT(2).EQ.BLANK) PNTR(I)=0
144:      120   CONTINUE
145:            GOTO 45
146:      900   CONTINUE
147:            RETURN
148:            END
```

	ATTRIBUTES	POSITION	SIZE
-- A2D --			
STATUS	INTEGER ARRAY	0	160.
PNTR	INTEGER ARRAY	13	96.
SCANS	REAL	11	
NODUB	LOGICAL	10	
INEX	INTEGER	7	
IDELT	INTEGER	6	
TIM2AD	INTEGER	5	
NXTBLK	INTEGER	4	
FILSIZ	INTEGER	3	
IOFF	INTEGER	2	
NNLST	INTEGER	1	
NEV	INTEGER	0	
EVNT	INTEGER ARRAY	240	160.
IPASS	INTEGER ARRAY	302	3.
SEC	REAL	300	
TIM	INTEGER ARRAY	273	7.
DUR	REAL	271	
DELT	REAL	267	
SIZEV	INTEGER	266	
IBLKEV	INTEGER	265	
NAM	INTEGER ARRAY	241	20.
EVNUM	INTEGER	240	
NAME	INTEGER ARRAY	323	16.
LIN	INTEGER ARRAY	300	80.
-- STATIC VARIABLES --			
BLANK	INTEGER	0	
-- STACK VARIABLES --			
ILIST	INTEGER	1	
ICHANGE	INTEGER	2	
I	INTEGER	3	
J	INTEGER	4	
IMODE	INTEGER	5	
NLENTD	INTEGER ARRAY	6	41.
NLENT	INTEGER ARRAY	6	40.

-- EXTERNAL SUBPROGRAMS --

.IACC	.FURS	.FRDI	.TACC	.NCAL
FDNAM	.IFUR	.TFUR	.FURI	.IEND
.IFRD	.TFRD	.SEEK	.IBUR	.BURI
.TBUR				

```
IDENTIFIER                    REFERENCES

ADERR       95   93
BLANK       19   128  142  143
BELT        98
BRLN        33   31
BRLNB       33   31
BUR         98
EVNT        55   56   21   98   114  137  142
EVNUM       93   98
EVNUMPS     94   93
FCA         94   93
FDNLPL      1
FDNAMES     114
FILSIZ      61   65
FILTLUN     35   31
FSSUV       95   93
GPSIZ       99   93
I           119  120  126  129  137  138  139  140  142  143  144
IBLKEV      98
IBLKPS      94
ICHANGE     115  116
IDELT       65
IREX        65
ILIST       109  110
INODE       113  114
IOFF        65
IPASS       93   98   114  137  142
J           127  120
LCA         94
LIN         56
NLENT       20   22   128  131  134  138  139  142  143
NLENTD      20   22
NLLN        34
NLLNB       34
NAM         93   98
NAME        20   21   138  139
NEV         65   100  114
NNLST       65   121  133  134
NODUB       32   65
NUMPASSES   35
NXTBLK      65
PNTR        61   65   120  140  143
PPSIZ       95   93
PSNUM       94   93
PSPNT       95   93
SCANS       65
SEC         98
SIZEV       93   98
SIZPS       94   93
STATUS      55   56   65
SWITCHLUN   35   31
TIM         93   98
TIM2AD      61   65
TSSUV       95   93

1           130  129
100         119
110         119  120
```

119	129	131	
120	126	144	
121	127	128	
2	132	131	
3	124	123	
40	109		
45	114	145	
50	100	110	115
900	116	146	

A2D 56

FDLOG.FR

```
      SUBROUTINE FDLOG
      INCLUDE "EVPARM.AD"
C*************************************************************************
C  PARAMETERS AND COMMON FOR THE EVCON PACKAGE
C
C
C
C***** GENERAL PARAMETERS *****
      INTEGER DRLN,DRLNB,SWITCHLUN,FILTLUN
      LOGICAL NODUB
      PARAMETER DRLN=160, DRLNB=DRLN*2
      PARAMETER NLLN=40, NLLNB=2*NLLN
      PARAMETER NUMPASSES=3,SWITCHLUN=6,FILTLUN=5
C  DRLN=LENGTH OF A RECORD IN THE EVENT DIRECTORY (EVDIR.AD)
C    DRLNB=THE SAME RECORD LENGTH AS DRLN, BUT IN BYTES
C  NLLN=LENGTH OF A RECORD IN THE MASTERLIST DIRECTORY (MASTERLIST.AD)
C    NLLNB= THE SAVE RECORD LENGTH AS NLLN, BUT IN BYTES
C    THE RECORD LENGTH IN BYTES IS USED WHEN OPENING THESE FILES
C  NUMPASSES=NUMBER OF PASSES NEEDED TO DIGITIZE ALL OF THE TAPE TRACKS
C
C       (EXPLAIN SET UP OF PASSES)
C
C  SWITCHLUN=LOGICAL UNIT NUMBER (LUN) OF THE OUTPUT BITS THAT CONTROL
C     THE TRACK SELECT SWITCH
C  FILTLUN=L.U.N. OF THE OUTPUT BITS THAT CONTROL THE FILTER SELECT SWITCH
C
C       (EXPLAIN SWITCH SELECTION)
C
C
C
C
C***** DEFINE COMMON AREA *****
      INTEGER STATUS,EVNT
      COMMON /A2D/ STATUS(DRLN),EVNT(DRLN),LIN(80)
C
C
C
C***** DEFINE STATUS ARRAY LAYOUT *****
      INTEGER PNTR(96),TIM2AD,FILSIZ
      EQUIVALENCE (NEV,STATUS(1)),(NMLST,STATUS(2)),(IOFF,STATUS(3)),
     *   (FILSIZ,STATUS(4)),(NXTBLK,STATUS(5)),(TIM2AD,STATUS(6)),(IDELT,STATUS(7)),
     *   (IHEX,STATUS(8)),(NODUB,STATUS(9)),(SCANS,STATUS(10)),
     *   (PNTR(1),STATUS(12))
C  THE FIRST RECORD IN THE DIRECTORY FILE (DPO:EVDIR) CONTAINS
C    GENERAL INFORMATION AND IS ALWAYS KEPT IN ARRAY "STATUS".
C    THE LAYOUT OF THIS RECORD IS AS FOLLOWS:
C ARRAY
C INDEX NAME     DESCRIPTION
C   1   NEV      NUMBER OF EVENTS ALREADY ON DISK
C   2   NMLST    NUMBER OF ENTRIES IN MASTERLIST FILE
C   3   IOFF     INITIAL OFFSET FOR CONTIGUOUS FILE "DPO:EVDAT"
C   4   FILSIZ   TOTAL USABLE SPACE IN "DPO:EVDAT", IN DISK BLOCKS
C   5   NXTBLK   NUMBER OF THE NEXT DISK BLOCK IN "DPO:EVDAT" TO START
C                  STORE DATA INTO (INITIALLY EQUALS "IOFF")
C   6   TIM2AD   THE NUMBER OF THE A/D CHANNEL RECIEVING A TIME CODE.
C                  0 IF NO TIME CODE
```

```
C    7   IDELT    TIME PARAMETER FOR P.I.T. THERE ARE IDELT*6.25
C                   MICROSECONDS BETWEEN SAMPLE POINTS
C    8   IMEX     FOOTAGE COUNT FOR TAPE POSITIONING FOR THE CURRENT EVENT
C    9   MODUS    TRUE IF THE TAPE BEING DIGITIZED IS NOT A DUBBED TAPE
C   10   SCANS    THE (REAL) NUMBER OF SAMPLES TO BE DIGITIZED ON THE
C                   CURRENT EVENT
C   12   PNTR     AN ARRAY USED TO MAP TAPE TRACK-CHANNEL TO A RECORD
C                   IN THE MASTERLIST FILE, WHICH IS SUPPOSED TO BE A
C                   STATION CARD IMAGE
C
C
C
C***** DEFINE EVNT ARRAY LAYOUT *****
      INTEGER FCA,GPSIZ,PPSIZ,ADERR,TSSUV,FSSUV,SIZPS,SIZEV,EVNUM,
     *   EVNUMPS,PSNUM,PSPNT,TIM(7),NAM(20),IPASS(NUMPASSES)
      PARAMETER EVNUMPS=0,PSNUM=1,IBLKPS=2,SIZPS=3,FCA=4,LCA=5
      PARAMETER PSPNT=6,TSSUV=38,FSSUV=39,ADERR=40,PPSIZ=ADERR+1
      EQUIVALENCE (EVNUM,EVNT(1)),(NAM(1),EVNT(2)),(IBLKEV,EVNT(22)),
     *   (SIZEV,EVNT(23)),(DELT,EVNT(24)),(DUR,EVNT(26)),
     *   (TIM(1),EVNT(28)),(SEC,EVNT(33)),(IPASS(1),EVNT(35))
      PARAMETER GPSIZ=38
C
C  <DISCRIBE SETUP>
C
C
C**********************************************************************
C
      EQUIVALENCE (ISPEED,EVNT(85)),(IRS,EVNT(86))
      JPAGE=5
C
C
C  CHECK IF THERE ARE ANY EVENTS TO LOG
      IF (NEV.GT.1) GOTO 20
            WRITE (10,7)
    7       FORMAT (/"  NO EVENTS ON THIS DISK"/)
            GOTO 900
C
   20     CONTINUE
      TYPE NEV-1," EVENTS ON THIS DISK"
   10 TYPE " "
C
      IF (NEV.EQ.1) GOTO 200
   30 ACCEPT "LOG WHICH EVENTS? (ENTER FIRST AND LAST): ",IF,IL
      IF ((IF.LT.1).OR.(IF.GT.IL).OR.(IL.GT.(NEV-1))) GOTO 30
      GOTO 210
C
  200   CONTINUE
            IF=1
            IL=1
  210 CONTINUE
      CALL CHRSIZ(4)
      WRITE (10,19)
   19 FORMAT ("(33)(14)")
      CALL FDELY(15)
C
      DO 110 J=IF,IL
            READ(1,REC=J+1) EVNT
            ITSF=ISPEED*IRS
            WRITE (10,12) J,(NAM(K),K=1,20)
   12       FORMAT ("EVENT",I4,7X,20S2)
```

```
                WRITE (10,3) TIM(2),TIM(3),TIM(1),TIM(4),TIM(5),SEC,IRS*16
  3             FORMAT (4X,"DATE: ",I2,"/",I2,"/",I2,14X,
     &            "TIME: ",I2,":",I2,":",F5.2,8X,"RECORDED AT 15/",I3," IPS")
                WRITE(10,5) DUR,IFIX(1./DELT+.5),ITSF
  5             FORMAT(4X,"DURATION:",F6.1," SECONDS ",3X,
     &            I4," SAMPLES PER SECOND",3X,"PLAYED BACK AT X ",I2)
 14             FORMAT (26A2/)
                        I=1
                        IFC=EVNT(IPASS(I)+FCA)
                        IF (IFC.EQ.0) GOTO 100
                        ILC=EVNT(IPASS(I)+LCA)
                        IMODE=1
                        CALL FDNAMES(J,I,EVNT(IPASS(I)+PSPNT),IFC,ILC,IMODE)
                        IFL=EVNT(IPASS(I)+FSSWV)
100             CONTINUE
101              CONTINUE
      IF ( (MOD((J-IF+1),JPAGE)).NE.0 ) GOTO 110
      TYPE "MAKE A HARD COPY IF DESIRED AND THEN HIT (CR)"
      READ (11,17) JPAUSE
 17   FORMAT ($1)
      WRITE (10,19)
      CALL FDELY(15)
110   CONTINUE
900   CONTINUE
      CALL CHRSIZ(3)
      RETURN
      END
```

	ATTRIBUTES	POSITION	SIZE
-- A2D --			
STATUS	INTEGER ARRAY	0	160.
PNTR	INTEGER ARRAY	13	96.
SCANS	REAL	11	
NODUB	LOGICAL	10	
INEX	INTEGER	7	
IDELT	INTEGER	6	
TIM2AD	INTEGER	5	
NXTBLK	INTEGER	4	
FILSIZ	INTEGER	3	
IOFF	INTEGER	2	
NNLST	INTEGER	1	
NEV	INTEGER	0	
EVNT	INTEGER ARRAY	240	160.
IRS	INTEGER	365	
ISPEED	INTEGER	364	
IPASS	INTEGER ARRAY	302	3.
SEC	REAL	300	
TIM	INTEGER ARRAY	273	7.
DUR	REAL	271	
DELT	REAL	267	
SIZEV	INTEGER	266	
IBLKEV	INTEGER	265	
NAM	INTEGER ARRAY	241	20.
EVNUM	INTEGER	240	
LIN	INTEGER ARRAY	300	80.
-- STACK VARIABLES --			
IF	INTEGER	1	
IL	INTEGER	2	
J	INTEGER	3	
K	INTEGER	4	
ITBF	INTEGER	5	
JPAUSE	INTEGER	6	
JPAGE	INTEGER	7	
I	INTEGER	10	
IFC	INTEGER	11	
ILC	INTEGER	12	
IMODE	INTEGER	13	
IFL	INTEGER	14	

-- EXTERNAL SUBPROGRAMS --

.IFWR	.TFWR	.ITYP	.FWRI	.FWRS
.TTYP	.IACC	.FRDI	.TACC	.NCAL
CNRSI	FDELY	.SEEK	.IBRD	.BRDI
.TBRD	.FWRR	FDNAM	.IFRD	.TFRD

IDENTIFIER	REFERENCES
ADERR	73 71
CHRSIZ	107 141
DELT	76 120
DRLN	11 9
DRLNB	11 9
DUR	76 120
EVNT	33 34 76 84 113 125 127 129 130
EVNUM	71 76
EVNUMPS	72 71
FCA	72 71
FDELY	110 138
FDLOG	1
FDNAMES	129
FILSIZ	39 43
FILTLUN	13 9
FSSUV	73 71
GPSIZ	77 71
I	124 125 127 129 130
IDLKEV	76
IDLKPS	72
IDELT	43
IF	99 100 104 112 133
IFC	125 126 129
IFL	130
INEX	43
IL	99 100 105 112
ILC	127 129
INODE	128 129
IOFF	43
IPASS	71 76 125 127 129 130
IRS	84 114 117
ISPEED	84 114
ITSF	114 120
J	112 113 115 129 133 139
JPAGE	85 133
JPAUSE	135
K	115
LCA	72
LIN	34
NLLN	12
NLLNB	12
NAM	71 76 115
NEV	43 89 95 98 100
NNLST	43
NODUB	10 43
NUMPASSES	13
NXTBLK	43
PNTR	39 43
PPSIZ	73 71
PSNUM	72 71
PSPNT	73 71
SCANB	43
SEC	76 117
SIZEV	71 76
SIZPS	72 71
STATUS	33 34 43
SWITCHLUN	13 9

FDDUMP.FR

```
1:    C                         TO MAG TAPE ON THE ECLIPSE
2:    C
3:    C       WRITTEN BY:       P. REASENBERG
4:    C                         JANUARY, 1980
5:    C
6:    C                         PROGRAM FDDUMP IS USED TO CREATE MAG TAPES
7:    C                         ON THE ECLIPSE WHICH CAN BE READ ON UNIX
8:    C
9:    C                         THE PROGRAM TRANSFERS SELECTED EVENTS FROM
10:   C                         FILES EVDAT AND EVDIR ON THE CURRENT
11:   C                         PORTABLE DISK PACK 'DP0' ON THE ECLIPSE.
12:   C
13:   C                         FILE DP0:EVDAT IS ASSUMED TO CONTAIN EVENT-
14:   C                         ORGANIZED MULTIPLEXED SEISMIC DATA FROM PROGRAM FDCON
15:   C                         FILE DP0:EVDIR IS AASSUMED TO CONTAIN THE ASSOCIATED
16:   C                         'DIRECTORY' INFORMATION ABOUT THAT EVENT.
17:   C
18:   C                         THIS PROGRAM, TOGETHER WITH PROGRAM 'FDCON' FORM THE COMPLETE
19:   C                         PACKAGE NEEDED TO GET 5-DAY TAPE DATA FROM ANALOG TAPES
20:   C                         TO UNIX-READABLE TAPES, VIA THE ECLIPSE
21:   C
22:   C
23:   C*************************************************************************
24:   C
25:
26:           INCLUDE "EVPARM.AD"
27:   C*************************************************************************
28:   C  PARAMETERS AND COMMON FOR THE EVCON PACKAGE
29:   C
30:   C
31:   C
32:   C***** GENERAL PARAMETERS *****
33:           INTEGER DRLN,DRLNB,SWITCHLUN,FILTLUN
34:           LOGICAL NODUB
35:           PARAMETER DRLN=160, DRLNB=DRLN*2
36:           PARAMETER MLLN=40, MLLNB=2*MLLN
37:           PARAMETER NUMPASSES=3,SWITCHLUN=6,FILTLUN=5
38:   C  DRLN=LENGTH OF A RECORD IN THE EVENT DIRECTORY (EVDIR.AD)
39:   C    DRLNB=THE SAME RECORD LENGTH AS DRLN, BUT IN BYTES
40:   C  MLLN=LENGTH OF A RECORD IN THE MASTERLIST DIRECTORY (MASTERLIST.AD)
41:   C    MLLNB= THE SAVE RECORD LENGTH AS MLLN, BUT IN BYTES
42:   C    THE RECORD LENGTH IN BYTES IS USED WHEN OPENING THESE FILES
43:   C  NUMPASSES=NUMBER OF PASSES NEEDED TO DIGITIZE ALL OF THE TAPE TRACKS
44:   C
45:   C       (EXPLAIN SET UP OF PASSES)
46:   C
47:   C  SWITCHLUN=LOGICAL UNIT NUMBER (LUN) OF THE OUTPUT BITS THAT CONTROL
48:   C     THE TRACK SELECT SWITCH
49:   C  FILTLUN=L.U.N.  OF THE OUTPUT BITS THAT CONTROL THE FILTER SELECT SWITCH
50:   C
51:   C       (EXPLAIN SWITCH SELECTION)
52:   C
53:   C
54:   C
55:   C
56:   C***** DEFINE COMMON AREA *****
```

```
57:           INTEGER STATUS,EVNT
58:           COMMON /A2D/ STATUS(DRLN),EVNT(DPLN),LIN(80)
59:     C
60:     C
61:     C
62:     C***** DEFINE STATUS ARRAY LAYOUT *****
63:           INTEGER PNTR(96),TIM2AD,FILSIZ
64:           EQUIVALENCE (NEV,STATUS(1)),(NMLST,STATUS(2)),(IOFF,STATUS(3)),
65:          *    (FILSIZ,STATUS(4)),(NXTBLK,STATUS(5)),(TIM2AD,STATUS(6)),(IDELT,STATUS(7)),
66:          *    (INEX,STATUS(8)),(NODUB,STATUS(9)),(SCANS,STATUS(10)),
67:          *    (PNTR(1),STATUS(12))
68:     C  THE FIRST RECORD IN THE DIRECTORY FILE (DPO:EVDIR) CONTAINS
69:     C     GENERAL INFORMATION AND IS ALWAYS KEPT IN ARRAY "STATUS"
70:     C     THE LAYOUT OF THIS RECORD IS AS FOLLOWS:
71:     C ARRAY
72:     C INDEX NAME    DESCRIPTION
73:     C   1   NEV     NUMBER OF EVENTS ALREADY ON DISK
74:     C   2   NMLST   NUMBER OF ENTRIES IN MASTERLIST FILE
75:     C   3   IOFF    INITIAL OFFSET FOR CONTIGUOUS FILE "DPO:EVDAT"
76:     C   4   FILSIZ  TOTAL USABLE SPACE IN "DPO:EVDAT", IN DISK BLOCKS
77:     C   5   NXTBLK  NUMBER OF THE NEXT DISK BLOCK IN "DPO:EVDAT" TO START
78:     C                 STORE DATA INTO (INITIALLY EQUALS "IOFF")
79:     C   6   TIM2AD  THE NUMBER OF THE A/D CHANNEL RECIEVING A TIME CODE
80:     C                 0 IF NO TIME CODE
81:     C   7   IDELT   TIME PARAMETER FOR P.I.T. THERE ARE IDELT*6 25
82:     C                 MICROSECONDS BETWEEN SAMPLE POINTS
83:     C   8   INEX    FOOTAGE COUNT FOR TAPE POSITIONING FOR THE CURRENT EVENT
84:     C   9   NODUB   TRUE IF THE TAPE BEING DIGITIZED IS NOT A DUBBED TAPE
85:     C  10   SCANS   THE (REAL) NUMBER OF SAMPLES TO BE DIGITIZED ON THE
86:     C                 CURRENT EVENT
87:     C  12   PNTR    AN ARRAY USED TO MAP TAPE TRACK-CHANNEL TO A RECORD
88:     C                 IN THE MASTERLIST FILE, WHICH IS SUPPOSED TO BE A
89:     C                 STATION CARD IMAGE
90:     C
91:     C
92:     C
93:     C***** DEFINE EVNT ARRAY LAYOUT *****
94:           INTEGER FCA,GPSIZ,PPSIZ,ADERR,TSSWV,FSSWV,SIZPS,SIZEV,EVNUM,
95:          *    EVNUMPS,PSNUM,PSPNT,TIM(7),NAM(20),IPASS(NUMPASSES)
96:           PARAMETER EVNUMPS=0,PSNUM=1,IBLKPS=2,SIZPS=3,FCA=4,LCA=5
97:           PARAMETER PSPNT=6,TSSWV=38,FSSWV=39 ADERR=40,PPSIZ=ADERR+1
98:           EQUIVALENCE (EVNUM,EVNT(1)),(NAM(1),EVNT(2)),(IBLKEV,EVNT(22)),
99:          *    (SIZEV,EVNT(23)),(DELT,EVNT(24)),(DUR,EVNT(26)),
100:         *    (TIM(1),EVNT(28)),(SEC,EVNT(33)),(IPASS(1),EVNT(35))
101:          PARAMETER GPSIZ=38
102:    C
103:    C  (DISCRIBE SETUP)
104:    C
105:    C
106:    C*************************************************************************
107:    C
108:    C
109:          DIMENSION NAME(16),LIST(104)
110:          INTEGER BUFF(4096),OUT(4096),EVENT_TRACE(1:STAT(:32),ZERO/0/
111:          INTEGER UID(7),TEMP(14),UNIT,FILE(4:TRIAN
112:          LOGICAL FIRST
113:          EQUIVALENCE (NAME(1),EVNT(52)) (IFF,TRACE(1:
114:    C
115:          PARAMETER NPASS=1, NTR=8, NBLK=256, IDCHN=7
116:    C
```

```
117:           OPEN 10,"$TT01"
118:           OPEN 11,"$TT11"
119:           WRITE (10,3)
120:         3 FORMAT ('(33)(14)')
121:           CALL FDELY(15)
122:     C
123:           CALL FGTIM(IHR,IMIN,ISEC)
124:           CALL FGDAY(IMO,IDAY,IYR)
125:           WRITE (10,1) IMO,IDAY,IYR,IHR,IMIN
126:         1 FORMAT (/'WELCOME TO PROGRAM FDDUMP !',5X,I2,'/',I2,'/',I2,5X,I2,':',I2,/
127:          &          /'THIS PROGRAM WILL DUMP MULTIPLEXED FIVE-DAY TAPE DATA'/
128:          &           'FROM DISK FILE DPO:EVDAT TO MAGNETIC TAPE IN '/
129:          &           'DEMULTIPLEXED (TRACE BY TRACE) FORMAT.'//
130:          &           'YOU CAN DUMP ALL EVENTS ON THE DISK FILE (BATCH MODE)'/
131:          &           'OR JUST SOME OF THEM (INTERACTIVE MODE)'//
132:          &           'YOU MAY CREATE EITHER BINARY RECORDS (4096 WORDS),'/
133:          &           'FOR USE ON UNIX,'/
134:          &           'OR ASCII FORMATTED (160015) RECORDS.'//
135:          &           'MOUNT A WRITE-ENABLED TAPE ON TAPE UNIT ZERO,'/
136:          &           'AND ANOTHER ON TAPE UNIT ONE IF YOU EXPECT TO'/
137:          &           'FILL UP THE FIRST TAPE.'//
138:          &           ' THEN STRIKE ANY KEY  .'/)
139:           CALL GCHAR (ITEST,IER)
140:     C
141:     C     SET UP TO READ EVDAT DATABASE
142:           CALL INIT("DPO",IER)
143:           IF (IER.NE.1.AND.IER.NE.40) GOTO 900
144:           CALL MCAMI
145:           CALL DIRC1
146:     C
147:     C     START WITH MAG TAPE UNIT MT0 AND LET
148:     C     "MTOUT" INIT AND OPEN IT THE FIRST TIME THROUGH
149:           ICODE = 0
150:           FIRST = .TRUE.
151:     C
152:     C     READ DPO:EVDIR, RECORD 1
153:           IREC=1
154:           READ (1,REC=IREC,ERR=905) STATUS
155:           WRITE (10,4) NEV
156:         4 FORMAT (/'THERE ARE ',I3,' EVENTS IN THE DATABASE ON THIS DISK'/)
157:     C
158:     C     CHOOSE PROGRAM MODE
159:       120 ACCEPT 'PROGRAM MODE (1=BATCH, 2=INTERACTIVE): ',MODE
160:     C
161:     C     CHOOSE TAPE FORMAT
162:       130 ACCEPT 'TAPE FORMAT  (1=BINARY (FOR UNIX), 2=ASCII): ',IFMT
163:           IF (IFMT.NE.1.AND.IFMT.NE.2) GOTO 130
164:     C
165:     C     SELECT EVENT(S)
166:           IF (MODE.EQ.1) GOTO 140
167:           IF (MODE.EQ.2) GOTO 145
168:           GOTO 120
169:       140 J1=1
170:           J2=NEV
171:           GOTO 150
172:       145 J1=1
173:           J2=1
174:     C
175:     C     ***  LOOP ON EVENTS - DUMP EACH TO TAPE  ***
176:       150 DO 1000 EVENT=J1,J2
```

```
177:            CALL RQEVT(LIST,EVENT)
178:    C
179:    C       READ EVENT INFO IN EVDIR AND PRINT IT ON TUBE
180:            IREC=EVENT+1
181:            READ (1,REC=IREC,ERR=905) EVNT
182:    C
183:    C       ENCODE  THE UNIQUE ID CONSISTING OF
184:    C       YYMMDDHHMMSSNN, WHERE NN = RECORDER NUMBER
185:    C
186:            ISEC=SEC
187:            ENCODE (UID,160) (TIM(K),K=1,5),ISEC,NAM(7)
188:      160   FORMAT (6I2,S2)
189:            DECODE (UID,161) (TEMP(K),K=1,14)
190:      161   FORMAT (14A1)
191:    C       CHANGE SPACES TO ZEROS
192:            DO 162 K=1,14
193:      162   IF (TEMP(K).EQ.'  ') TEMP(K)='0 '
194:            ENCODE (UID,161) (TEMP(K),K=1,14)
195:    C
196:    C       WRITE EVENT INFO TO TUBE
197:            WRITE (10,6) EVENT,(NAM(K),K=1,20),TIM(2),TIM(3),TIM(1),
198:           &          TIM(4),TIM(5),SEC,DUR,IFIX(1./DELT+.5)
199:         6  FORMAT (//'EVENT',I4,7X,20S2,/
200:           &          'DATE: ',I2,'/',I2,'/',I2,14X,'TIME: ',I2,':',I2,':',F5.2,/
201:           &          4X,'DURATION: ',F6.1,' SECONDS ',3X,I4,' SAMPLES PER SECOND')
202:            WRITE (10,163) (UID(K),K=1,7),SIZEV
203:      163   FORMAT ('     UNIQUE ID: ',7A2,5X,I5,' DISK BLOCKS')
204:    C
205:            IF (MODE.EQ.1) GOTO 200
206:    C
207:    C       INTERACTIVE CHECK ON EVENTS
208:      155   WRITE (10,156)
209:      156   FORMAT ('--> STRIKE (CR) TO DUMP THIS EVENT, (SPACE) TO SKIP: ',Z)
210:            CALL GCHAR (ITEST,IER)
211:            IF (ITEST.EQ.015K) GOTO 200
212:            IF (ITEST.EQ.040K) GOTO 1000
213:            GOTO 155
214:    C
215:    C       DETERMINE DPTS, THE NUMBER OF DATA VALUES IN EACH TRACE
216:      200   DPTS = SIZEV*FLOAT(NBLK/NTR)
217:            WRITE (10,201)
218:      201   FORMAT (' <--')
219:    C
220:    C       FOR BINARY TAPES, WRITE 4096-WORD (8192-BYTE) RECORDS
221:    C       FOR ASCII  TAPES, WRITE 4000-WORD (8000-BYTE) RECORDS
222:            IF (IFMT.EQ.1) LREC=4096
223:            IF (IFMT.EQ.2) LREC=4000
224:    C
225:    C       DETERMINE NREC, THE NUMBER OF TAPE RECORDS NEEDED FOR DUMPING A TRACE
226:            NREC = (DPTS+(LREC-1))/LREC
227:    C
228:    C       DETERMINE THE STARTING RECORD NUMBERS FOR EACH TRACE (RELATIVE
229:    C       TO THE LAST E-O-F)   FIRST RECORD IN FILE IS RECORD 1
230:            DO 210 ITR=1,NTR
231:      210   ISTART(ITR) = (ITR-1)*NREC+2
232:    C
233:    C       ***  GENERATE HEADER RECORD  ***
234:    C
235:    C       LOAD NAME OF EVENT
236:            DO 221 J=1,20
```

```
      221   BUFF(J)=EVNT(J+1)
C
C       MULTIPLY DELT BY 100000. AND LOAD INTO TWO INTEGER WORDS
        XDELT=DELT*100000.+.5
        BUFF(21)=IFIX(XDELT/32768.)
        BUFF(22)=AMOD(XDELT,32768.)
C
C       MULTIPLY DUR BY 100. AND LOAD INTO TWO INTEGER WORDS
        XDUR=DUR*100.+.5
        BUFF(23)=IFIX(XDUR/32768.)
        BUFF(24)=AMOD(XDUR,32768.)
C
C       LOAD YR,MO,DAY,HR,MIN
        DO 222 J=25,29
      222   BUFF(J)=EVNT(J+3)
C
C       MULTIPLY SEC BY 1000. AND LOAD INTO TWO INTEGER WORDS
        XSEC=SEC*1000.+.5
        BUFF(30)=IFIX(XSEC/32768.)
        BUFF(31)=AMOD(XSEC,32768.)
C
C       LOAD TRACE NAMES
        DO 223 J=32,47
      223   BUFF(J)=EVNT(J+20)
C
C       LOAD UNIQUE ID
        DO 230 K=1,7
      230   BUFF(47+K) = UID(K)
C
C       LOAD STARTING RECORD NUMBERS OF TRACES (ISTART)
        DO 231 K=1,8
      231   BUFF(54+K) = ISTART(K)
C
C
C       LOAD DPTS INTO TWO INTEGER WORDS
        BUFF(63)=IFIX((DPTS+.5)/32768.)
        BUFF(64)=AMOD((DPTS+.5),32768.)
C
C       ZERO OUT THE REST OF THE HEADER BUFFER
        DO 224 J=65,LREC
      224   BUFF(J)=0
C
        IF (IFMT EQ.2) GOTO 235
C
C       BINARY FORMAT FOR HEADER
        CALL FORM1(BUFF,OUT)
        GOTO 240
C
C       ASCII FORMAT FOR HEADER
      235   ENCODE (OUT,236) (BUFF(J),J=1,20),DELT,DUR,(EVNT(J),J=26,32),
     &                    SEC,(EVNT(I),J=52,67),(EVNT(I),J=1,7),
     &                    (ISTART(J),J=1,8) DPTS
      236   FORMAT (20A2,2F10.3,5I5,F10.3,16A2,4I2,I3,I5,F10.3,8000X)
C
C       WRITE THE HEADER RECORD
      240   ICOM =1
        IF (FIRST) ICOM=0
        CALL MIOUT (IPCHN,ICOM,LREC,OUT,ICODE,IER)
        FIRST = .FALSE.
        IF (IER EQ 7) GOTO 240
```

```
297:            IF (IER NE.1) GOTO 925
298:    C
299:    C       ***  GENERATE TRACE DATA RECORDS  ***
300:    C
301:    C       LOOP ON PASSES
302:      300   DO 350 IP=1,NPASS
303:            CALL RQPAZ(STA,EVENT,IP)
304:    C
305:    C       LOOP ON TRACES
306:      400   DO 470 ITR=1,NTR
307:    C
308:    C       SET UP FRAME SIZE
309:            CALL FRAMD (LREC,TRACE,1,1,IER)
310:            IF (IER.NE.0) GOTO 920
311:    C
312:    C       GET A FRAME
313:      405   CALL GETFM(BUFF,NUMBR,IEND)
314:            IF (NUMBR EQ.LREC) GOTO 430
315:    C
316:    C       ZERO OUT REST OF BUFFER
317:            DO 420 I=NUMBR+1,LREC
318:      420   BUFF(I)=0
319:      430   IF (IFMT.EQ.2) GOTO 435
320:    C
321:    C       BINARY FORMAT
322:            CALL FORM1(BUFF,OUT)
323:            GOTO 440
324:    C
325:    C       ASCII 1600I3 FORMAT
326:      435   CALL FORM2(BUFF,OUT)
327:    C
328:    C       WRITE A TRACE RECORD
329:      440   ICOM=1
330:            CALL MTOUT (TPCHN,ICOM,LREC,OUT,ICODE,IER)
331:            IF (IER.EQ.7) GOTO 240  ;RE-WRITE THIS FILE ON THE NEXT TAPE UNIT
332:            IF (IER.NE.1) GOTO 925
333:            IF (IEND.EQ.0) GOTO 405
334:    C
335:    C       GET THE NEXT TRACE
336:      470   CONTINUE
337:    C
338:    C       GET THE NEXT PASS
339:      350   CONTINUE
340:    C
341:    C       WRITE AN E-O-F AT END OF EVENT
342:            CALL FDELY(10)
343:            ICOM=2
344:            CALL MTOUT (TPCHN,ICOM,IDUM,OUT,ICODE,IER)
345:            IF (IER NE 1) GOTO 925
346:            IF (MODE EQ.1) GOTO 1000
347:     1000   CONTINUE
348:    C
349:    C       ***  NORMAL ENDING  ***
350:            IF (MODE EQ 2) GOTO 1010
351:    C
352:    C       END OF BATCH JOB
353:            WRITE (10,1001)
354:     1001   FORMAT (/'END OF BATCH JOB')
355:            CLOSE 3
356:            CALL RESET
```

```
357:           CALL RELEASE ("MT1",IER)
358:           CALL RELEASE ("MT0",IER)
359:           STOP
360:   C
361:    1010   J1=J1+1
362:           J2=J2+1
363:           IF (J1.LE.NEV) GOTO 150
364:   C
365:   C       END OF INTERACTIVE JOB
366:           WRITE (10,1011)
367:    1011   FORMAT (//'NO MORE EVENTS ON DISK FILE')
368:           CLOSE 3
369:           CALL RESET
370:           CALL RELEASE ("MT1",IER)
371:           CALL RELEASE ("MT0",IER)
372:           STOP
373:   C
374:   C       ***  ABNORMAL ENDINGS  ***
375:     900   WRITE (10,901)
376:     901   FORMAT ('ERROR - CANNOT INITIALIZE DIRECTOPY DPO')
377:           CALL RESET
378:           STOP
379:   C
380:     902   WRITE (10,903)
381:     903   FORMAT ('ERROR IN OPENING DPO:EVDIF')
382:           CALL RESET
383:           STOP
384:   C
385:     905   WRITE (10,906) IREC
386:     906   FORMAT ('READ ERROR IN DPO:EVDIR - RECORD = ',I3/
387:        &          'PREVIOUSLY DUMPED EVENTS ON THIS TAPE ARE O K ')
388:           CALL RESET
389:           STOP
390:   C
391:     908   WRITE (10,909)
392:     909   FORMAT ('ERROR IN OPENING DPO:MASTERLIST')
393:           CALL RESET
394:           STOP
395:   C
396:   C
397:     910   WRITE (10,911) IER
398:     911   FORMAT ('ERROR - CANNOT INITIALIZE MAG TAPE UNIT - IER= ',I5,/
399:        &          'CHECK DEVICE AND START AGAIN')
400:           CALL RESET
401:           STOP
402:   C
403:     920   WRITE (10,921) EVENT,IP,ITR,IER
404:     921   FORMAT ('FRAMD ERROR - EVENT = ',I3,'  PASS=',I2,
405:        &          'TRACE=',I3,'  IER=',I3,/
406:        &          'PREVIOUSLY DUMPED EVENTS ON THIS TAPE ARE O K ')
407:           CALL RESET
408:           STOP
409:   C
410:     925   WRITE (10,926) EVENT,IP,ITR,IER,ICOM,ICODE
411:     926   FORMAT ('MTOUT ERROR - EVENT=',I3,'  PASS=',I2,'  TRACE=',I3,'  IER=',I3/
412:        &          'ICOM = ',I2,' ICODE = ',O6,'K')
413:           CALL RESET
414:           STOP
415:   C
416:           END
```

ATTRIBUTES		POSITION	SIZE
-- A2D --			
STATUS	INTEGER ARRAY	0	160
PNTR	INTEGER APRAY	13	96.
SCANS	REAL	11	
NODUB	LOGICAL	10	
IHEX	INTEGER	7	
IDELT	INTEGER	6	
TIM2AD	INTEGER	5	
NXTBLK	INTEGER	4	
FILSIZ	INTEGER	3	
IOFF	INTEGER	2	
NMLST	INTEGER	1	
NEV	INTEGER	0	
EVNT	INTEGER APRAY	240	160.
NAME	INTEGER ARRAY	323	16.
IPASS	INTEGER ARRAY	302	3.
SEC	REAL	300	
TIM	INTEGER ARRAY	273	7.
DUR	REAL	271	
DELT	REAL	267	
SIZEV	INTEGER	266	
IBLKEV	INTEGER	265	
NAM	INTEGER ARRAY	241	20.
EVNUM	INTEGER	240	
LIN	INTEGER ARRAY	500	80.
-- STATIC VARIABLES --			
ZERO	INTEGER	0	
-- STACK VARIABLES --			
EVENT	INTEGER	1	
UNIT	INTEGER	2	
FIRST	LOGICAL	3	
IMO	INTEGER	4	
IDAY	INTEGER	5	
IYP	INTEGER	6	
IHR	INTEGER	7	
IMIN	INTEGER	10	
MODE	INTEGER	11	
IFMT	INTEGER	12	
J1	INTEGER	13	
J2	INTEGER	14	
K	INTEGER	15	
ISEC	INTEGER	16	
NREC	INTEGER	17	
J	INTEGER	20	
LREC	INTEGER	21	
IP	INTEGER	22	
I	INTEGER	23	
NUMBR	INTEGER	24	
IREC	INTEGER	25	

IER	INTEGER	26	
ICOM	INTEGER	27	
ICODE	INTEGER	30	
ITEST	INTEGER	31	
IEND	INTEGER	32	
IDUM	INTEGER	33	
BUFF	INTEGER ARRAY	34	4096
OUT	INTEGER ARRAY	10034	4096.
TRACE	INTEGER ARRAY	20034	1.
ITR	INTEGER	20034	
STA	INTEGER ARRAY	20035	64.
UID	INTEGER ARRAY	20135	7.
TEMP	INTEGER ARRAY	20144	14.
FILE	INTEGER ARRAY	20162	4.
LIST	INTEGER ARRAY	20166	104.
DPTS	REAL	20336	
XDELT	REAL	20340	
XDUR	REAL	20342	
XSEC	REAL	20344	

-- EXTERNAL SUBPROGRAMS --

.FOPE	.IFVR	.TFVR	.MCAL	FDELY
FGTIM	FGDAY	.FVPI	GCHAR	INIT
MCAM1	D1PC1	.IERR	.SEEK	IBRD
.BRDI	.TBRD	.IACC	.FVRS	.FRDI
.TACC	RBEVT	.IENC	.TENC	.IDEC
.TDEC	.FVRR	AMOD	FORM1	MTOUT
RBPAZ	FRAMD	GETFM	FORM2	FCLO
RESET	RELEA	.STOP		

```
IDENTIFIER                      REFERENCES

ADERR       97   95
AMOD        242  247  256  273
BUFF        110  237  241  242  246  247  251  255  256  260  264  268  272  273
            277  282  288  313  318  322  326
DELT        100  198  240  288
DIRC1       145
DPTS        216  226  272  273  288
DRLN        35   33
DRLNB       35   33
DUR         100  198  245  288
EVENT       110  176  177  180  198  303  347  403  410
EVNT        57   58   100  113  181  237  251  260  288
EVNUM       95   100
EVNUMPS     96   95
FCA         96   95
FDELY       121  342
FGDAY       124
FGTIM       123
FILE        111
FILSIZ      63   67
FILTLUN     37   33
FIRST       112  150  293  295
FORM1       282  322
FORM2       326
FRAND       309
FSSWV       97   95
GCHAR       139  210
GETFM       313
GPSIZ       101  95
I           317  318
IBLKEV      100
IBLKPS      96
ICODE       149  294  330  344  410
ICOM        292  293  294  329  330  343  344  410
IDAY        124  125
IDELT       67
IDUM        344
IEND        313  333
IER         139  142  143  210  294  296  297  309  310  330  331  332  344  345
            357  358  370  371  397  403  410
IFMT        162  163  222  223  279  319
IHEX        67
IHR         123  125
IMIN        123  125
IMO         124  125
INIT        142
IOFF        67
IP          302  303  339  403  410
IPASS       95   100
IREC        153  154  180  181  385
ISEC        123  186  187
ITEST       139  210  211  212
ITF         113  230  231  306  336  403  410
IYR         124  125
J           236  237  250  251  259  260  276  277  288
J1          169  172  176  361  363
J2          170  173  176  362
```

```
150       171  176  363
155       208  213
156       209  208
160       188  187
161       190  189  194
162       192  193
163       203  202
200       205  211  216
201       218  217
210       230  231
221       236  237
222       250  251
223       259  260
224       276  277
230       263  264
231       267  268
235       279  288
236       289  288
240       283  292  296  331
3         120  119
300       302
350       302  339
4         156  155
400       306
405       313  333
420       317  318
430       314  319
435       319  326
440       323  329
470       306  336
6         201  198
900       143  375
901       376  375
902       380
903       381  380
905       154  181  385
906       387  385
908       391
909       392  391
910       397
911       399  397
920       310  403
921       406  403
925       297  332  345  410
926       412  410

A2D       58
```

MTOUT.FR

```
      SUBROUTINE MTOUT (TPCHN,CON,LREC,ARRAY,ICODE,IER)
C
C
C     WRITTEN BY:     P. REASENBERG
C                     U.S. GEOLOGICAL SURVEY
C
C     DATE:           JANUARY, 1980
C
C
C     GENERAL PURPOSE SUBROUTINE FOR WRITING TO MAG TAPES,
C     USING DATA GENERAL ROUTINE "MTDIO".
C
C
C
C     TPCHN   INPUT PARAMETER DESIGNATING FORTRAN TAPE CHANNEL FOR MAG TAPE UNIT.
C
C     LREC    INPUT PARAMETER SPECIFYING NUMBER OF INTEGER WORDS TO BE
C             TRANSFERED TO TAPE IN ONE RECORD
C
C     CON     INPUT PARAMETER SPECIFYING ACTION ROUTINE WILL TAKE:
C
C             CON = 0 INIT MAG TAPE UNIT NUMBER PASSED IN "ICODE".
C                     POSITION TAPE TO END OF LAST EXISTING FILE ON TAPE.
C                     WRITE A DATA RECORD (LREC WORDS) TO THE TAPE.
C
C             CON = 1 ASSUMES TAPE UNIT (NUMBER PASSED IN "ICODE") IS INITIALIZED.
C                     ASSUMES TAPE IS CORRECTLY POSITIONED.
C                     WRITE A DATA RECORD (LREC WORDS) TO THE TAPE.
C
C             CON = 2 ASSUMES TAPE UNIT (NUMBER PASSED IN "ICODE") IS INITIALIZED.
C                     ASSUMES TAPE IS CORRECTLY POSITIONED.
C                     WRITES AN END-OF-FILE MARK ON THE TAPE.
C
C             NOTE:   FIRST CALL TO "MTOUT" MUST BE WITH CON=0. SUBSEQUENT CALLS
C                     MUST BE WITH EITHER CON=1 OR CON=2.
C
C
C     ARRAY   INPUT ARRAY CONTAINING THE DATA TO BE WRITTEN TO TAPE.
C             MUST BE 4096 INTEGERS OR LESS.
C
C     ICODE   UPON INPUT:
C             ICODE = 0       BEGIN WITH TAPE UNIT "MT0"
C             ICODE = 1       BEGIN WITH TAPE UNIT "MT1"
C
C             UPON OUTPUT (IER=1,2,3,4,6,7)
C             ICODE = 0 OR 1, INDICATING THE MAG TAPE UNIT TO USE IN THE
C                     NEXT CALL TO "MTOUT"
C
C             UPON OUTPUT (IER=5)
C             ICODE = TAPE STATUS WORD (TSW) RETURNED BY "MTDIO"
C
C
C     IER     OUTPUT PARAMETER RETURNED BY SUBROUTINE INDICATING
C             STATUS OF RETURN:
C
C                      IER = 1          NORMAL RETURN
```

```
C                       IER = 2         INVALID VALUE FOR "COM"
C                       IER = 3         INVALID VALUE FOR "LREC"
C                       IER = 4         INVALID VALUE FOR "ICODE"
C                       IER = 5         UNKNOWN ERROR RETURNED BY "MTDIO"
C                       IER = 6         CANT INITIALIZE TAPE UNIT.
C                       IER = 7         PHYSICAL END OF TAPE ENCOUNTERED,
C                                       OR 100 FILES ON TAPE ALREADY.
C                                       CURRENT RECORD AND FILE NOT WRITTEN TO TAPE.
C                                       REWRITE CURRENT FILE ON UNIT SPECIFIED BY
C                                       VALUE RETURNED IN "ICODE".
C                       IER = 8         CANNOT OPEN TAPE FILE VIA MTEND
C
C
C**********************************************************************
C
      INTEGER ARRAY(1),FILE(4),COM,STATUS,TPCHN,UNIT
      DATA FILE/'MT','0 ','00',0/
C
C     CHECK CALLING PARAMETERS FOR VALIDITY
      IF (COM.EQ.1.OR.COM.EQ.0) GOTO 5
      IF (COM.EQ.2) GOTO 6
      IER=2
      RETURN
    5 IF (LREC.GE.2.AND.LREC.LE.4096) GOTO 6
      IER=3
      RETURN
    6 IF (ICODE.EQ.0.OR.ICODE.EQ.1) GOTO 7
      IER=4
      RETURN
C
C     SELECT TAPE UNIT
    7 UNIT=ICODE
      IF (UNIT.EQ.0) TUNIT='MT0 '
      IF (UNIT.EQ.1) TUNIT='MT1 '
C
      GOTO (50,100,200) COM+1
C
C     OPEN INITIAL TAPE UNIT AND POSITION TAPE TO END IF THE
C     TAPE HAS FILES. OTHERWISE, POSITION TAPE TO BEGINNING.
   50 CALL INIT (TUNIT,0,IER)
      IF (IER.NE.1.AND.IER.NE.40) GOTO 900
      CALL MTEND (TPCHN,TUNIT,FILE,IER)
      IF (IER.EQ.-1) GOTO 500
      IF (IER.EQ.2)  GOTO 905
      WRITE (10,51) FILE
   51 FORMAT (/'*** WRITING TO TAPE FILE ',4A2,' ***')
C
C
C
C     WRITE A DATA RECORD OF LREC WORDS
  100 ICMD=50000K .OR. MOD(LREC,4096)
      IERR=1
      CALL MTDIO (TPCHN,ICMD,ARRAY,STATUS,IERR)
      IF (IERR.NE.1) GOTO 150
      ICODE=UNIT
      IER=1
      RETURN
C
  150 IF (IERR.EQ.28) GOTO 160
  155 TYPE 'MTOUT: MTDIO RETURNS IERR = ',IERR
```

```
117:            IER=5
118:            ICODE=STATUS
119:            RETURN
120:
121:      160   IF ((STATUS.AND.1000K).EQ.0) GOTO 155
122:            GOTO 500      ; PHYSICAL END OF TAPE ENCOUNTERED
123:    C
124:    C       WRITE AN END-OF-FILE MARK
125:      200   ICMD = 60000K    ; E-O-F
126:            CALL MTDIO (TPCHN,ICMD,0,STATUS,IERR)
127:            CALL MTDIO (TPCHN,ICMD,0,STATUS,IERR)
128:            ICMD=40001K    ; BACKSPACE
129:            CALL MTDIO (TPCHN,ICMD,0,STATUS,IERR)
130:            IF (IERR.NE.20) GOTO 155
131:            IER=1
132:            ICODE=UNIT
133:            RETURN
134:    C
135:    C       END OF THE TAPE ENCOUNTERED ...OR... 100 FILES ALREADY ON THIS TAPE
136:    C       BACKSPACE TO END OF LAST FILE AND WRITE SOME E-O-F MARKS
137:      500   ICMD=40000K        ;BACKSPACE TO BEGINNING OF LAST FILE
138:            CALL MTDIO (TPCHN,ICMD,0,STATUS,IERR)
139:            ICMD=30000K        ;FOWARD TO BEGINNING OF NEXT (INCOMPLETE) FILE
140:            CALL MTDIO (TPCHN,ICMD,0,STATUS,IERR)
141:            ICMD=60000K      ;E-O-F
142:            CALL MTDIO (TPCHN,ICMD,0,STATUS,IERR)
143:            CALL MTDIO (TPCHN,ICMD,0,STATUS,IERR)
144:            ICMD=40001K    ; BACKSPACE
145:            CALL MTDIO (TPCHN,ICMD,0,STATUS,IERR)
146:            IF (IERR.NE.20) GOTO 155
147:    C
148:            CLOSE TPCHN
149:            CALL RELEASE (TUNIT,IER)
150:            WRITE (10,512) TUNIT
151:      512   FORMAT (/'*** FINISHED WITH TAPE UNIT ',A4,'***')
152:    C
153:    C       SWITCH TO THE OTHER TAPE UNIT
154:            IF (TUNIT.EQ."MTO ") GOTO 510
155:            TUNIT="MTO "
156:            WRITE (10,511) TUNIT
157:            UNIT=0
158:            FILE(2)="0:"
159:            GOTO 515
160:      510   TUNIT="MT1 "
161:            WRITE (10,511) TUNIT
162:      511   FORMAT (/'*** SWITCHING TO TAPE UNIT ',A4,'***')
163:            UNIT=1
164:            FILE(2)="1:"
165:      515   CALL INIT (TUNIT,IER)
166:            IF (IER.NE.1.AND.IER.NE.40) GOTO 900
167:    C
168:    C       OPEN NEW TAPE UNIT AND POSITION TAPE TO END OF LAST FILE
169:            CALL MTEND (TPCHN,TUNIT,FILE,IER)
170:            IF (IER EQ.-1) GOTO 500     ; ALREADY 100 FILES ON THIS TAPE
171:            IF (IER EQ 2) GOTO 905
172:            WRITE (10,51) FILE
173:            ICODE=UNIT
174:            IER=7
175:            RETURN
176:    C
```

```
177:    C
178:    C                ERROR AND ABNORMAL RETURN HANDLING
179:      900    ICODE=UNIT
180:             TYPE 'MTOUT: INIT RETURNS IER = ',IER
181:             IER=6
182:             RETURN
183:    C
184:      905    ICODE=UNIT
185:             IER=0
186:             RETURN
187:    C
188:             END
```

```
            ATTRIBUTES          POSITION  SIZE

-- DUMMY ARGUMENTS --

TPCHN     INTEGER                 @-3
COM       INTEGER                 @-4
LREC      INTEGER                 @-5
ARRAY     INTEGER ARRAY           @-6
ICODE     INTEGER                 @-7
IER       INTEGER                 @-10


-- STATIC VARIABLES --

FILE      INTEGER ARRAY           0         4.


-- STACK VARIABLES --

STATUS    INTEGER                 1
UNIT      INTEGER                 2
IERR      INTEGER                 3
ICMD      INTEGER                 4
TUNIT     REAL                    5


-- EXTERNAL SUBPROGRAMS --

.CGB              .MCAL             MNIT              MTEND             .IFWR
.FWRI             .TFWR             MTDIO             .ITYP             .FWRS
.TTYP             .FCLO             RELEA             .FWRR
```

IDENTIFIER	REFERENCES													
ARRAY	72	1	109											
CON	72	1	76	77	92									
FILE	72	73	98	101	158	164	169	172						
ICMD	107	109	125	126	127	128	129	137	138	139	140	141	142	143
	144	145												
ICODE	1	83	88	111	118	132	173	179	184					
IER	1	78	81	84	96	97	98	99	100	112	117	131	149	165
	166	169	170	171	174	180	181	185						
IERR	108	109	110	115	116	126	127	129	130	138	140	142	143	145
	146													
INIT	96	165												
LREC	1	80	107											
MTDIO	109	126	127	129	138	140	142	143	145					
MTEND	98	169												
MTOUT	1													
RELEASE	149													
STATUS	72	109	118	121	126	127	129	138	140	142	143	145		
TPCHN	72	1	98	109	126	127	129	138	140	142	143	145	148	169
TUNIT	89	90	96	98	149	150	154	155	156	160	161	165	169	
UNIT	72	88	89	90	111	132	157	163	173	179	184			
100	92	107												
150	110	115												
155	116	121	130	146										
160	115	121												
200	92	125												
5	76	80												
50	92	96												
500	99	122	137	170										
51	102	101	172											
510	154	160												
511	162	156	161											
512	151	150												
515	159	165												
6	77	80	83											
7	83	88												
900	97	166	179											
905	100	171	184											

MTEND.FR

```
 1:           SUBROUTINE MTEND (FORCHN,UNIT,FILE,IERR)
 2:   C
 3:   C       TO GET TO THE END OF THE SPECIFIED MAG TAPE UNIT
 4:   C       AND TO OPEN THAT UNIT AT THE BEGINNING OF THE NEXT FILE.
 5:   C
 6:   C       INPUTS:
 7:   C               FORCHN -        FORTRAN CHANNEL DESIGNATED FOR TAPE UNIT
 8:   C
 9:   C               UNIT -          REAL VARIABLE CONTAINING THE ASCII
10:   C                               STRING 'MT0 ' OR 'MT1 '
11:   C       OUTPUTS:
12:   C               FILE            FOUR-INTEGER ARRAY CONTAINING AN ASCII
13:   C                               STRING REPRESENTING CURRENT OPENED UNIT:FILE
14:   C                               FOR EXAMPLE, 'MT','0:','14','  '
15:   C
16:   C               IERR            =1  ALL OK
17:   C                               =-1  ALREADY 100 FILES ON THIS TAPE. DO NOT OPEN.
18:   C                               =2   CANNOT OPEN THE TAPE FILE
19:   C
20:   C
21:   C*************************************************************************
22:   C
23:           INTEGER FILE(4),FORCHN,STATUS,DUM(10)
24:   C
25:           DO 10 J=0,99
26:           IF (J.GE.10) GOTO 2
27:           ENCODE (FILE,1) UNIT,J
28:        1  FORMAT (A3,':0',I1,2X)
29:           GOTO 4
30:        2  ENCODE (FILE,3) UNIT,J
31:        3  FORMAT (A3,':',I2,2X)
32:        4  FILE(4)=0
33:           CALL OPEN (FORCHN, FILE, 3, IER)
34:           IF (IER.EQ.13) GOTO 15
35:           IF (IER.NE.1)  GOTO 11
36:           CALL CLOSE (FORCHN,IER)
37:       10  CONTINUE
38:       11  IERR=-1
39:           RETURN
40:   C
41:       15  IF (J.NE.2) GOTO 16
42:   C       SEE IF THIS IS AN INIT/F TAPE OR NOT
43:           CALL MTDIO (FORCHN,10000K,0,STATUS,IERR) ; REWIND TAPE
44:           CALL MTDIO (FORCHN,10,DUM,STATUS,ICNT,IERR) ; READ TEN WORDS FROM FIRST FILE
45:           IF (ICNT.EQ.10) GOTO 16  ; NOT AN INIT/F'ED TAPE
46:           J=0
47:           GOTO 17
48:       16  J=J-1
49:       17  IF (J.GE.10) GOTO 20
50:           ENCODE (FILE,1) UNIT,J
51:           GOTO 25
52:       20  ENCODE (FILE,3) UNIT,J
53:       25  FILE(4)=0
54:           CALL MTOPD (FORCHN,FILE,0,IER)
55:           IF (IER.NE.1) GOTO 30
56:           IERR=1
```

```
57:           RETURN
58:      30   IERR=2
59:           RETURN
60:           END
```

	ATTRIBUTES	POSITION	SIZE

-- DUMMY ARGUMENTS --

FORCHN	INTEGER	0-3	
UNIT	REAL	0-4	
FILE	INTEGER ARRAY	0-5	
IERR	INTEGER	0-6	

-- STACK VARIABLES --

STATUS	INTEGER	1	
J	INTEGER	2	
IER	INTEGER	3	
ICNT	INTEGER	4	
DUM	INTEGER ARRAY	5	10.

-- EXTERNAL SUBPROGRAMS --

.IENC	.FURR	.FURI	.TENC	.NCAL
OPEN	CLOSE	MTDIO	MTOPD	

IDENTIFIER	REFERENCES										
CLOSE	36										
DUM	23	44									
FILE	23	1	27	30	32	33	50	52	53	54	
FORCHN	23	1	33	36	43	44	54				
ICNT	44	45									
IER	33	34	35	36	54	55					
IERR	1	38	43	44	56	58					
J	25	26	27	30	37	41	46	48	49	50	52
NTDIO	43	44									
NTEND	1										
NTOPD	54										
OPEN	33										
STATUS	23	43	44								
UNIT	1	27	30	50	52						
1	28	27	50								
10	25	37									
11	35	38									
15	34	41									
16	41	45	48								
17	47	49									
2	26	30									
20	49	52									
25	51	53									
3	31	30	52								
30	55	58									
4	29	32									

ECLIPSE FORTRAN 5, VERSION 5.21 -- THURSDAY, DECEMBER 27, 1979 2:49:39 PM

FORM1.FR

```
1:            SUBROUTINE FORM1 (IN,OUT)
2:            DIMENSION IN(4096)
3:            INTEGER OUT(4096)
4:            DO 1 K=1,4096
5:        1   OUT(K)=IN(K)
6:            RETURN
7:            END
```

FORM2.FR

```
1:
2:            SUBROUTINE FORM2 (IN,OUT)
3:            DIMENSION IN(1600)
4:            INTEGER OUT(4000)
5:            ENCODE (OUT,1) (IN(K),K=1,1600)
6:        1   FORMAT (1600I5)
7:            RETURN
8:            END
```

```
--------fdread.f--------Mon Apr 14 11:25:59 PST 1980--------

c       program fdread - to read 5-day archive tapes into UNIX files
c
c
c       Written by                          Paul Reasenberg
c                                              Jan, 1980
c
c       Modified for UNIX(tm) by            Jeff Hobson
c                                              Mar, 1980
c
c
c
        implicit integer*2 (i-n)
        parameter (tpchn=3, lrec=4096, lhead=72, ntr=8)
        double precision time
        integer*2    head(lhead),data(lrec),tpchn,tnam(ntr * 2),eof
        integer*4    len
        character*40 name
        character*14 uidl(100),uid,ofile(ntr + 1)
        character*4  tname(ntr),itest
        character    string(35)
        dimension    istart(ntr),id(7),nam(20)
        equivalence  (nam(1),name),(id(1),uid),(tnam(1),tname(1))
        equivalence  (nam(1),head(1)), (idelt1,head(21)),
     &          (idelt2,head(22)), (idur1,head(23)), (idur2,head(24)),
     &          (iyr,head(25)), (imo,head(26)), (iday1,head(27)),
     &          (ihr,head(28)), (imin,head(29)), (isec,head(30)),
     &          (isec2,head(31)), (tnam(1),head(32)), (id(1),head(48)),
     &          (istart(1),head(55)),(idpts1,head(63)),(idpts2,head(64))
c
c       determine program mode

        mrec = 0
        open (8,file="fdread.out")
        time = clock(8)
        call datstr(time,2,-8.,string,len,2)
        write(8,'(35a1)') string
 10     len = 0
        call tpread(len,data,mrec,ier)
        write (6,11)string
 11     format (/,35a1,//,'          PROGRAM MODES: ',/,
     &          '1 - load all events on tape',/,
     &          '2 - interactively select events to load',/,
     &          '3 - load only events in a list',/,
     &          "4 - no load -  make a list of event id's on tape",/,
     &          "5 - stop the program",/)
 12     call printn('enter mode please ')
        read (5,'(i)',err=12) mode
        if (mode.lt.1.or.mode.gt.5) call printn('Invalid response; re-')
        if (mode.lt.1.or.mode.gt.5) go to 12
        if (mode.eq.5) go to 460
        if (mode.ne.3) go to 60
c
c       generate the list of unique id's to load
 20     call printn("\nEnter the 14-character unique id's you want")
        call printn('\n one per line.  Enter an empty line to quit\n')
        j=0
 22     j=j+1
 23     read (5,'(a)',err=26) uidl(j)
        if (uidl(j).eq."              ") go to 33
        if (j.gt.100) go to 30
        go to 22
```

```
 63  26     call printn("please re-enter last unique id: ")
 64         go to 23
 65
 66  30     call printn("too many events ")
 67  33     nev=j-1
 68         call printn("%d events will be loaded",nev)
 69 c
 70 c       open tape file for reading at file 0
 71  60     open (tpchn,file ='/dev/mt0',status = 'old',recl = lrec)
 72         rewind tpchn
 73         close(tpchn)
 74 c
 75 c
 76 c       read record 1 (lhead words) into array head
 77  70     call tpread (lhead,head,mrec,ier)
 78 c
 79 c               0 = EOF
 80 c               - = ERR
 81 c               + = OK
 82 c
 83         if (ier.gt.0) go to 71
 84 c           .
 85 c       process tape error condition
 86 c
 87         if(ier.eq.0) go to 420
 88         call printn("header error %d\n",ier)
 89         go to 400
 90
 91 c       convert from header integers to character variables
 92 c
 93 c       reconstruct real variables from integer-coded variables on tape
 94  71     eof = 0
 95         do 73 i=21,31
 96         call inswap(head(i))
 97  73     continue
 98         do 74 i=55,64
 99         call inswap(head(i))
100  74     continue
101         delt=(idelt1*32768.+idelt2)/100000.
102         dur =(idur1 *32768.+idur2 )/100.
103         sec =(isec1 *32768.+isec2 )/1000.
104         dpts=(idpts*32768.+idpts2)
105 c
106         if (mode.eq.1) go to 120
107         if (mode.eq.2.or.mode.eq.4) go to 100
108 c
109 c       search list of uid's for current uid
110  90     do 95 k=1,nev
111         if (uid.ne.uid1(k)) go to 95
112 c
113 c               found a match !!!!
114 c
115
116         go to 120
117
118  95     continue
119         go to 400
120 c
121 c       print description of event to terminal
122  100    call printn('\nevent id %s - %s\n date: %d/%02d/%d',
123 &       uid,name,imo,iday1,iyr)
124         call printn('              time: %d:%02d:%05.2f\n',ihr,imin,sec)
125         call printn('    duration: %.1f seconds     %n samples/sec\n',
```

92

```
c
      if (mode.eq.4) go to 400
c
c     check to see if this event is wanted
 110  call printn('\nDo you want to load this event? ')
      read(5,'(a)')itest
      if (itest.eq. "yes") go to 120
      if (itest.eq. "no" ) go to 400
      if (itest.eq. "NO" ) go to 460
      if (itest.eq. "quit") go to 460
      if (itest.eq. "menu") go to  10
      go to 110
c
c     create file names
 120  open(9,file='/tmp/reasen',status='scratch')
      do 121 j = 1,ntr
 121  write(9,'(a14,i1)') uid,j
      write(9,'(a14,"h")') uid
      rewind 9
      do 122 j = 1,ntr+1
 122  read(9,'(1x,a14)')ofile(j)
      close(9,status='delete')
c
c
c     open for writing the header file and write header
 130  open (9, file=ofile(ntr+1),form='formatted',status='new')
c
      write (9,101) uid,name,imo,iday1,iyr,ihr,imin,sec,dur,
     &              ifix(1./delt+0.5)
      write (9,132) (k,tname(k),k=1,ntr)
 132  format(4(5x,i1,' = ',a4))
      close (9)
c
c     read and write the trace data. Tape file is assumed
c     to be positioned at beginning of record 2.
c
c     determine number of records to read for each trace
      last = istart(2) - istart(1)
c
      do 150 itr=1,ntr
      open (9,file=ofile(itr),status='new')
c
      do 145 jrec=1,last
      call tpread (lrec,data,mrec,ier)
      if(ier.gt.0) go to 142
      if(ier.eq.0) go to 410
      call printn("data error %d\n",ier)
      go to 400
c
 142  npts=lrec
      if (jrec.eq.last) npts=dpts-(last-1)*lrec
      do 143 i=1,npts
      call inswap(data(i))
 143  continue
 145  write (9,'(16i5)') (data(k),k=1,npts)
c
      close (9)
 150  continue
c
      write (6,'(a14," loaded")') uid
      write (8,101) uid,name,iyr,imo,iday,ihr,imin,sec,dur,
     &              ifix(1./delt+0.5)
 101  format(/,'event id ',a16,' - ',a40,/,
```

```
191 &       'time: ',i2,':',i2,':',f5.2,/,
192 &       4x,'duration: ',f6.1,' seconds ',i3,' samples/second')
193         go to 70
194 c
195 c       code to skip the current event and reposition the tape to
196 c       the beginning of the next file (event). If physical or
197 c       logical end of tape is encountered, go to 420
198 c
199  400    call tpread(lrec,data,mrec,ier)
200         if(ier.gt.0) go to 400
201         if(ier.eq.0) go to 420
202         call printn("skip error mrec = %d ier = %d\n",mrec,ier)
203         close (tpchn)
204         go to 10
205 c
206  410    close (9)
207         call printn('EOF on data read\n')
208         call printn('Last file written was %s %d of %d records\n',
209 &                   ofile(itr),jrec,last)
210  420    eof = eof + 1
211         mrec = 0
212         if(eof.lt.2) go to 70
213  450    call printn (" no more events on this tape")
214         close (9)
215         close (8)
216         close (tpchn)
217         go to 10
218
219  460    stop
220         end
```

```
--------tpread.c--------Mon Apr 14 11:26:06 PST 1980--------

 1 /*      Subroutine to read a DGC tape written @ 800 BPI and 4096
 2       words/block.
 3
 4               USE:    call tpread(len,buff,mrec,ier)
 5
 6               WHERE:  len = length of buffer
 7                       buff= integer buffer to receive data
 8                       mrec= number of records read so far
 9                               updated by this program
10                       ier = error return from read
11                               0 = E.O.F.
12                               - = ERROR
13                               + = O.K.
14
15               AUTHOR: Jeff Hobson
16                       United States Geological Survey
17                       Menlo Park, Calif
18
19               DATE:   Feb. 21, 1980
20
21 */
22
23      tpread_(len,buff,mrec,ier) long *len;int *mrec,*ier;char buff[];
24 {
25 #include <errno.h>
26      extern errno;
27      int i,in,no;
28      static fi=1234;
29      char ibuf[8192];
        ----------------------------------------

30      errno=0;
31      if(*len==0){
32              if(fi!=1234){
33                      close(fi);fi=1234;}return;}
34      if(fi==1234){
35              if((fi=open("/dev/nrmt0",0))<0)
36                      {*ier= -errno;fi=1234;perror("open error");}}
37      if((*ier=read(fi,ibuf,8192))<0)
38              {*ier= -errno;close(fi);fi=1234;perror("read error");}
39      if(*ier==0)
40              {(*mrec)++;return;}
41      no= (*len)*2;
42      if(*ier<no)
43              {no= *ier;printf("re-define # of bytes %d\n",no);}
44      for(i=0;i<no;i++)
45              buff[i]=ibuf[i];
46      (*mrec)++;return;
47 }
```

--------swap.c--------Mon Apr 14 11:26:11 PST 1980--------

```
 1 /*      Subroutine to swap the two bytes of a D.E.C.
 2         integer.  The primary use of this routine is
 3         to "patch-up" an integer read from an Eclipse
 4         tape.
 5
 6                 Author  Jeff Hobson
 7                         U.S.G.S
 8                         Menlo Park, Calif
 9
10                 Date    March, 1980
11
12                 Use     call inswap(i)
13                    where:
14                         i = integer to have its bytes swapped.
15                                 (this is done in-place!)
16
17 */
18
19 inswap_(w)char w[];{char c;c=w[0];w[0]=w[1];w[1]=c;}
```