UNITED STATES DEPARTMENT OF THE INTERIOR

U.S. GEOLOGICAL SURVEY

Documentation and Users Guide for USRFILES:

DISCO Seismic Data Base Information Retrieval Program

by

F.N. Zihlman

Open File Report 88-355

[1]U.S. Geological Survey, Box 25046, MS 960, Denver Federal Center, Denver, CO 80225

1988

CONTENTS

ILLUSTRATIONS

APPENDIX

Documentation and Users Guide for USRFILES: DISCO Seismic Data Base Information Retrieval Program

By F.N. Zihlman

ABSTRACT

When the DISCO seismic data processing system was installed at the USGS facility in Denver, CO, users had no method for retrieving information contained in the system's data base. USRFILES was written to provide a quick and convenient interactive method of searching the DISCO release 6.5 (or earlier) data base for and listing information about magnetic tape output data sets, velocity models and filter files. With information obtained by means of interactive prompts, USRFILES will search the specified seismic project MODELDIR.MDL file for the appropriate .USR file, open that file and display the desired information. DISCO release 7.0 totally restructured the seismic data base, replacing the MODELDIR.MDL with SDBDIR.IDX and .USR files with .SDB files, and provided a library of FORTRAN subroutines to be used to access the new structure. A new version of USRFILES was produced to access the new structure using the FORTRAN subroutines provided. Any information obtained by USRFILES may be displayed to the terminal screen, sent to a disk file, or both. All versions of USRFILES were written in DEC FORTRAN 77 for use in the VAX VMS environment.

INTRODUCTION

In 1982 the U.S. Geological Survey began processing seismic data using the newly acquired Digicon DISCO processing system on a VAX 11/780 computer. At that time there was no convenient way to access and list the contents of the seismic data base created and maintained by the DISCO system. This database contains information about seismic line geometry, magnetic tapes written during data processing, filters and velocity models. Users had to log on the VAX, go to the proper subdirectory in the seismic data base containing their DISCO project, search the project index file (MODELDIR.MDL) for the name of the .USR file containing the information about a specific processing sequence, and list the contents of that file in order to obtain information like the output tape reel identification, data set name, starting and ending ensemble numbers, etc. USRFILES, written in DEC FORTRAN 77, was originally produced to provide a quick and convenient way to search the MODELDIR.MDL for the proper .USR file describing an output data set, open that file, and list the desired information. Later versions were modified to list information about velocity and filter files in the data base.

With the advent of DISCO version 7.0, CogniSeis (formerly Digicon) completely reorganized the seismic data base structure. SDBDIR.IDX replaced the MODELDIR.MDL of the old data base and .SDB files replaced .USR files. With the new data base, CogniSeis provided both a library of FORTRAN subroutines, allowing programmers access to the new data base structure and a set of seismic data base utilities. These utilities, provided by CogniSeis to DISCO users for the first time, allowed interactive access to items contained in the seismic data base. However, none of these utilities provided the same functions of USRFILES with

either the ease of use or the summary of data base information returned. Users were faced with many of the same data base access problems which prompted the original development of USRFILES. As a result a new version of USRFILES was produced to access the new data base structure using the subroutine libraries provided by CogniSeis. The source code for USRFILES using those subroutine libraries is listed in the appendix.

PROGRAM OVERVIEW AND USER PROMPTS

USRFILES is started on the Geophysics Group's VAX by typing USRFILES at the VMS prompt. This executes a DCL command procedure which, in turn, will execute the appropriate FORTRAN version of USRFILES. If the terminal or terminal emulator is a device type the VMS operating system recognizes as a DEC terminal the command procedure will query the user as to which type of DISCO data base to access: DISCO version 6.5 or earlier (OLD), or DISCO version 7.0 or later (NEW). USRFILES version 4.1 (V4.1) is used to access the OLD seismic data base and USRFILES version 6.1 (V6.1) the NEW seismic data base. Both V4.1 and V6.1 use DEC terminal screen management features to display information to the user. If the user is using a terminal or terminal emulator which does not look like a DEC terminal to the VMS operating system a generic type (not using any screen management features) USRFILES (V4.0) will be initiated allowing access to the OLD data base only. The program dialog presented to the user in V4.0 is similar to that described for V4.1 below. Default options to program prompts are displayed to the user in square brackets and may be taken by pressing the carriage return in response to the prompt. All versions of USRFILES are case sensitive and expect all responses to any prompts to be in upper case only. Entering "EXIT" to any prompt will terminate the program and return the user to the VMS operating system. Figure 1 shows the on-screen display presented the user on a VT100-type terminal in response to typing USRFILES to the VMS prompt.

PROGRAM OUTPUT DIRECTION

Upon selecting either the OLD or NEW seismic data base, both V4.1 and V6.1 query the user for the program output direction (V4.0 does this last). The data base information may be displayed to the terminal screen, stored on disk, or both. Information stored on disk is sent to a file named SEISDATA_INFO.LIS located in the user's login directory. Figure 2 shows the output direction prompt displayed to the user for V4.1. V6.1 is similar.

SEISDATA PROJECT

After selection of the output direction the user is prompted for the seismic project name. If the project name does not exist, V4.1 returns a nasty looking FORTRAN run-time routine error message and returns the user to the DCL prompt for selecting the OLD or NEW data base. If the project name is entered incorrectly a second time, V4.1 again displays the nasty error message, exits the program and returns the user to the VMS operating system prompt. Both V6.1 and V4.0 politely inform the user the project can not be found. V6.1 will continuously prompt for the correct project name while V4.0 returns the user to the VMS prompt. Figure 3 shows the seismic project prompt for V4.1. V6.1 is similar.

```
                Seismic Project Data Base Information Pgm

                CONTROL_Y HALTS/REINITIALIZES PROGRAM

terminal type:  VT100

To access the OLD seismic data base (PRE-DISCO 7.0).....enter: OLD
To access the NEW seismic data base (DISCO 7.0).........enter: NEW


enter OLD, NEW, exit.....[OLD]:
```

Figure 1.--Data base selection display on a VT100 terminal.

```
                Seismic Data Base Information Pgm V4.1

Seismic data base information may be displayed to either the terminal screen,
a file on disk which may be saved and/or printed, or both.

enter SCREEN, FILE, or BOTH.....[SCREEN]:
```

Figure 2.--Output direction prompt displayed for V4.1.

```
                Seismic Data Base Information Pgm V4.1

All default responses are shown in brackets: [DEFAULT].
Default response taken by hitting <CARRIAGE RETURN>.
To stop program execution, enter EXIT to any prompt.
-------------------------------------------------------------------------------

enter <SEISDATA PROJECT>, exit.....[EXIT]:
```

Figure 3.--Seismic project prompt for V4.1

DATA TYPE

With the successful entry of the seismic project name, the user is prompted for the type of data base information to retrieve. V4.0, V4.1 and V6.1 can retrieve information about DISCO magnetic tape data sets (SET), velocity models (VEL) or filter files (FLT). V6.1 can also retrieve information about geometry models (GEOM). The user is continuously prompted for the data type to retrieve until one of the available types or "exit" is entered. Figures 4a and 4b show the data type prompt for V4.1 and V6.1.

DATA TYPE SET

If the data type selected is SET, versions V4.0, V4.1 and V6.1 will prompt the user for the line identification and data set name to use in the search process. V6.1 will further prompt the user for the version number of the data set. V4.1 and V6.1 allow full wild card usage, as in VMS, for both the line identification and data set name. V6.1 also allows wild card usage in the version number. V4.0 only allows either the exact line identification to use in the search or "ALL" to allow data sets of all lines in the data base to be searched. V4.0 does allow wild card usage on the right side only of the data set name entered or all data sets may be searched for and displayed by again entering "ALL". The "ALL" function in V4.1 and V6.1 is performed by entering a "*" in response to the prompt. Figures 5a and 5b show V4.1 line identification and data set name prompts. Figures 5c, 5d, and 5e show the V6.1 line identification, data set name and version number prompts.

DATA TYPE VEL

If the data type selected is VEL, versions V4.0, V4.1 and V6.1 will prompt the user for the line identification, as discussed for data type SET, and the velocity identification. As described above for the data set name, V4.1 and V6.1 will accept wild cards in the velocity identification while V4.0 will accept wild cards on the right side only, or all velocity models may be searched for and displayed by entering "ALL". The "ALL" function for V4.1 and V6.1 is performed by entering "*". Figures 6a and 6b show the V4.1 line and velocity identification prompts. V6.1 is similar.

DATA TYPE FLT

If the data type selected is FLT, V4.0, V4.1 and V6.1 will prompt the user for the line identification followed by the filter identification. All rules discussed for entering the SET and VEL information apply to the data type FLT as well. Figures 7a and 7b show the V4.1 line and filter identification prompts. V6.1 is similar.

DATA TYPE GEOM

V6.1 is the only version of USRFILES to return information on line geometry pattern models in the seismic data base. If the data type selected is GEOM, V6.1 will prompt the user for the pattern number to be used in the search. As with the other V6.1 prompts wild cards may be used. Figure 8 shows the V6.1 geometry prompt for the pattern number.

```
                Seismic Data Base Information Pgm V4.1

Project: MATTICK

FILE TYPE:                ENTER:
-------------             ------
data set...................SET
velocity file..............VEL
filter file................FLT


enter SET, VEL, FLT, exit.....[EXIT]:
```

Figure 4a.--Data type prompt for V4.1

```
                Seismic Data Base Information Pgm V6.1

Project: NPRA

FILE TYPE:                ENTER:
-------------             ------
data set...................SET
velocity file..............VEL
geometry file..............GEOM
filter file................FLT


enter SET, VEL, GEOM, FLT, exit.....[EXIT]:
```

Figure 4b.--Data type prompt for V6.1

```
Seismic Data Base Information Pgm V4.1

Project: MATTICK                Data type: SET

enter <LINE ID>, *, exit.....[*        ]:
```

Figure 5a.--Line Identification prompt for V4.1 data type SET.

```
Seismic Data Base Information Pgm V4.1

Project: MATTICK            line: GMT1

data type selected: SET

enter <DATA SET NAME>, *, exit.....[*                ]:
```

Figure 5b.--Data set name prompt for V4.1 data type SET.

```
                Seismic Data Base Information Pgm V6.1

Project: NMGSI2                 Data type: SET

enter <LINE ID>, *, exit.....[*       ]:
```

Figure 5c.--Line identification prompt for V6.1 data type SET.

```
                Seismic Data Base Information Pgm V6.1

Project: NMGSI2              line: L24

data type selected: SET

enter <DATA SET NAME>, *, exit.....[*              ]:
```

Figure 5d.--Data set name prmpt for V6.1 data type SET.

```
                Seismic Data Base Information Pgm V6.1

Project: NMGSI2              line: L24

data type selected: SET                  set name: *DMUX*

enter <VERSION NUMBER>, *, exit.....[*   ]:
```

Figure 5e: Version number prompt for V6.1 data type SET.

```
Seismic Data Base Information Pgm V4.1

Project: MATTICK                Data type: VEL

enter <LINE ID>, *, exit.....[*        ]:
```

Figure 6a.--Line identification prompt for V4.1 data type VEL.

```
Seismic Data Base Information Pgm V4.1

Project: MATTICK            line: GMT1

data type selected: VEL

enter <VELOCITY ID>, *, exit.....[*                ]:
```

Figure 6b.--Velocity identification prompt for V4.1 data type VEL.

```
Seismic Data Base Information Pgm V4.1

Project: MATTICK                Data type: FLT

enter <LINE ID>, *, exit.....[*        ]:
```

Figure 7a.--Line identification prompt for V4.1 data type FLT.

```
Seismic Data Base Information Pgm V4.1

Project: MATTICK            line: GMT1

data type selected: FLT

enter <FILTER ID>, *, exit.....[*                ]:
```

Figure 7b.--Filter identification prompt for V4.1 data type FLT.

```
Seismic Data Base Information Pgm V4.1

Project: NMGSI2                 line: GEOMETRY

data type selected: GEOMETRY

enter PATTERN NUMBER, *, exit.....[*       ]:
```

Figure 8.--Geometry pattern number prompt for V6.1 data type GEOM.

USRFILES INFORMATION RETURNED

V4.0, V4.1 and V6.1 will return information found in the seismic project database for which identification parameters match those entered by the user. The listings continue until no other matches can be made. If the data output is directed to the screen V4.1 and V6.1 will display only one screen at a time, then stop and wait for the user to press the carriage return. V4.0 will scroll information off the screen if the unsuspecting user does not stop it.

Data type SET will return the line identification, data set version number, data set name, seismic data base .USR file name for V4.1 and .SBD file name for V6.1, the .USR or .SDB file creation date, output tape density, data sort order, sample rate in milliseconds, number of samples, data length in milliseconds, and all tape reels associated with the data set. Tape reel information includes the tape reel identification, starting ensemble and trace number, ending ensemble and trace number, and the total number of traces on each reel. V6.1 will additionally return a data set description if the processor has assigned one. Figures 9a and 9b show the display format and information returned for data type SET for V4.1 and V6.1 respectively.

Data type VEL will return the line identification, velocity identification and the data base .USR (V4.1) or .SDB (V6.1) file name. V6.1 will additionally return a data set description if one has been assigned and the creation date of that file. Figures 10a and 10b show the display format and information returned for V4.1 and V6.1 respectively.

Data type FLT will return the line identification, filter identification and the data base .USR (V4.1) of .SDB (V6.1) file name. V6.1 will additionally return a data set description if one has been assigned and the creation date of that file. Figures 11a and 11b shows the display format and information returned for V4.1 and V6.1 respectively.

Data type GEOM (V6.1 only) will return the geometry pattern number, the .SBD file name and the creation date of that file. Figure 12 shows the display format and information returned.

INSTALLATION REQUIREMENTS

All versions of USRFILES were written in DEC FORTRAN 77, using DEC FORTRAN language extensions. If they should need to be recompiled DEC's FORTRAN compiler should be used. V4.0 was written using standard FORTRAN output statements. V4.1 was written using screen management run-time libraries no longer supported by DEC, which may cause problems if it needs to be recompiled. V6.1 was written using the current screen management routines and should compile. The CogniSeis seismic data base subroutines need to be added to V6.1 when the source code is compiled and linked as shown below:

```
$FORTRAN {source code}+{directory path}DCIFORDEF.TLB/LIB

$LINK {source object},{directory path}SDBOPTLIB/LIB,SYS$INPUT/OPTIONS
-{directory path}DCIOSI/SHAREABLE
```

```
Seismic Data Base Information Pgm V4.1

line: GMT1          version: 02     set: GMT1_DMUX

USR File: DCI_SDBDSK:[SEISDATA.MATTICK]M000002FF.USR
date: 5-JUL-1985  06:29:46.29

density:    6250     sort order: SHOT
sample rate (MS):  4.000     samples:    3250     data length (MS): 13000

                     first                   last                    total
       reel          ensemble/trace          ensemble/trace          traces
   1)  768                   1     1               502    14         12038
   2)  948                 502    15               998    21         11911
   3)  1704                998    22              1494    14         11897
   4)  919                1494    15              1740    24         5914

hit <CARRIAGE RETURN> to continue
```

Figure 9a.--Display format and information returned for V4.1 data type SET.

```
                Seismic Data Base Information Pgm V6.1

line: L24          version: 01     set: L24_DMUX

description:

SEISD$DISK:[SEISDATA.NMGSI2]T224.SDB;1
date: 8-DEC-1987  09:46:21.32

density:    6250     sort order: SHOT
sample rate (MS):  4.000     samples:    4000     data length (MS): 16000

                  first                    last                    total
     reel         ensemble/trace           ensemble/trace          traces
 1)  274                1     1                202    35          9683
 2)  275              202     36               412    42          10087
 3)  276              412     43               440    34          1336

hit <CARRIAGE RETURN> to continue
```

Figure 9b.--Display format and information returned for V6.1 data type SET.

```
Seismic Data Base Information Pgm V4.1

line: GMT1          velocity id: GMT1BR1
USR File: DCI_SDBDSK:[SEISDATA.MATTICK]M00000231.USR

hit <CARRIAGE RETURN> to continue
```

Figure 10a.--Display format and information returned for V4.1 data type VEL.

```
Seismic Data Base Information Pgm V6.1

line: L24           version: VELDEFN      set: WFA_VEL

description:

SEISD$DISK:[SEISDATA.NMGSI2]T231.SDB;1
date: 8-DEC-1987  09:47:01.62

hit <CARRIAGE RETURN> to continue
```

Figure 10b.--Display format and information returned for V6.1 data type VEL.

```
Seismic Data Base Information Pgm V4.1

line: L1            filter id: L1WF0
USR File: DCI_SDBDSK:[SEISDATA.MATTICK]M000002DC.USR

hit <CARRIAGE RETURN> to continue
```

Figure 11a.--Display format and information returned for V4.1 data type FLT.

```
Seismic Data Base Information Pgm V6.1

line: L24           version: FILTER22      set: YOMAMA

description:

SEISD$DISK:[SEISDATA.NMGSI2]T269.SDB;1
date: 25-DEC-1987  04:15:01.62

hit <CARRIAGE RETURN> to continue
```

Figure 11b.--Display format and information returned for V6.1 data type VEL.

```
Seismic Data Base Information Pgm V6.1

GEOMETRY      PATTERN       pattern: 0001

description:
GEOMETRY PATTERN DEFINITION

SEISD$DISK:[SEISDATA.NMGSI2]T312.SDB;1
date: 8-DEC-1987  09:54:49.05

hit <CARRIAGE RETURN> to continue
```

Figure 12.--Display format and information returned for V6.1 data type GEOM.

Currently V4.1 is coded to look for the DISCO SEISDATA directory in the root directory of a device having the logical name SEISD$DISK. If the seismic data base must be placed on another device with a different logical name USRFILES can be made to find that logical device through the use of the VMS ASSIGN command by:

$ASSIGN NEW_LOGICAL_NAME SEISD$DISK

This command may be placed in the command procedure which runs V4.1 or in the VAX system startup file.

Similarly, V4.0 is coded to look for a device having a logical name DCI_SDBDSK. As above, a logical name assignment using the VMS command ASSIGN will allow USRFILES to access any device desired.

V6.1 uses the CogniSeis subroutine SDB_SDBOPN using the current logical name definitions on the computer system to locate the specified project files.

PROGRAM AVAILABILITY

The USRFILES command procedure, V4.0, V4.1 and V6.1 source code and executable images are available in VAX VMS backup format on request from:

Nick Zihlman
U.S. Geological Survey
P.O. Box 25046, Mail Stop 960
Denver, CO 80225-0046
303-236-5741

Please provide a 9 track recording tape and specify either 1600 or 6250 bpi.

APPENDIX

```
C ------USRFILES_MAIN: used to examine Ver 7.0 DISCO seismic data base files
C       written by F.N. Zihlman, APR. 1986
C
C     subroutines used:
C     -----------------
C     USRFILES_SET:  selects/displays data base files created using TAPOUT
C                     module.
C
C     USRFILES_VEL:  selects/displays the name of the velocity file and the
C                     velocity id created by a DEFINE module (USRFILES_SETDSP).
C
C     USRFILES_GEOM: selects/displays GEOMETRY file names (USRFILES_GEODSP).
C
C     USRFILES_FLT:  selects/displays filter files selected (USRFILES_FLTDSP).
C
C     Each subroutine displays the data it finds in it's particular format
C
C     USRFILES_FLOW: sets output direction to SCREEN, FILE or BOTH
C ==========================================================================
C
C     VAR'S:
      LOGICAL       PROJECTLG,DATYPELG,EOPGM,OPLOG
C
      STRUCTURE /UINFO/
         CHARACTER*56 KEYID
         CHARACTER*8  PROJECT,LINEID
         CHARACTER*6  FLOW
         CHARACTER*4  VERNO,DATYPE
      END STRUCTURE
C
      STRUCTURE /SINFO/
         CHARACTER*80 DESCRIP,FILENAME
         CHARACTER*56 KEY
         CHARACTER*24 SDDATE
         CHARACTER*8  LINE,CLASS
      END STRUCTURE
C
      STRUCTURE /DINFO/
         INTEGER*4 PID
         INTEGER*4 KBID
         INTEGER*4 VID1
         INTEGER*4 VID2
      END STRUCTURE
C
      RECORD /UINFO/USRINFO
      RECORD /SINFO/SEISINFO
      RECORD /DINFO/DIDS
C
      CHARACTER*78 STR,TEXT,ERRMSG
      CHARACTER*8  PROJECT,LINEID
      CHARACTER*6  FLOW
      CHARACTER*4  DATYPE,DUMMY
```

```
C
      INTEGER*4 SMG$CREATE_PASTEBOARD,SMG$CREATE_VIRTUAL_DISPLAY
      INTEGER*4 SMG$PASTE_VIRTUAL_DISPLAY,SMG$PUT_CHARS
      INTEGER*4 SMG$PUT_LINE,SMG$CREATE_VIRTUAL_KEYBOARD
      INTEGER*4 SMG$BEGIN_DISPLAY_UPDATE,SMG$END_DISPLAY_UPDATE
      INTEGER*4 SMG$ERASE_DISPLAY,SMG$HOME_CURSOR,SMG$ERASE_CHARS
      INTEGER*4 SMG$PUT_WITH_SCROLL,SMG$READ_STRING
      INTEGER*4 SMG$RING_BELL,SMG$SET_CURSOR_ABS,SMG$DRAW_LINE
C
      COMMON /UFO/USRINFO
      COMMON /SFO/SEISINFO
      COMMON /DFO/DIDS
C
C ------include files necessary for return status codes, etc.
C          using the DIGICON seisdata routines
C
      INCLUDE '(OSIINC)'
      INCLUDE '(SDBDEF)'
      INCLUDE '($SMGDEF)'
      INCLUDE '($SSDEF)'
      INCLUDE '($LIBDEF)'
C
C

C ---------------------------------------------------------------------
C
      EOPGM=.FALSE.
      PROJECTLG=EOPGM
      DATYPELG=EOPGM
      OPLOG=EOPGM
      USRINFO.PROJECT='EXIT'
      USRINFO.DATYPE='EXIT'
      USRINFO.LINEID='*'
      USRINFO.KEYID='*'
      USRINFO.VERNO='*'
      USRINFO.FLOW='SCREEN'

C
C ------------------------------------------------------------
C     INITIALIZE DISPLAYS
C ------------------------------------------------------------
C ------create virtual pasteboard, virtual_id1, virtual_id2, virtual keyboard
C
      ISTAT=SMG$CREATE_PASTEBOARD(DIDS.PID,,IROW,ICOL,)
      IF(.NOT.ISTAT)THEN
        ERRMSG='SMG$CREATE_PASTEBOARD'
        GOTO 9000
      ENDIF
C
      ISTAT=SMG$CREATE_VIRTUAL_DISPLAY(1,78,DIDS.VID1,SMG$M_BORDER
      1,,)
```

```
      IF(.NOT.ISTAT)THEN
        ERRMSG='SMG$CREATE_VIRTUAL_DISPLAY (VID1)'
        GOTO 9000
      ENDIF
C
      ISTAT=SMG$CREATE_VIRTUAL_DISPLAY(20,78,DIDS.VID2,,,)
      IF(.NOT.ISTAT)THEN
        ERRMSG='SMG$CREATE_VIRTUAL_DISPLAY (VID2)'
        GOTO 9000
      ENDIF
C
      ISTAT=SMG$CREATE_VIRTUAL_KEYBOARD(DIDS.KBID)
      IF(.NOT.ISTAT)THEN
        ERRMSG='SMG$CREATE_VIRTUAL_KEYBOARD (KBID)'
        GOTO 9000
      ENDIF

C
C ------paste virtual displays-----------------------------------------
C
      ISTAT=SMG$PASTE_VIRTUAL_DISPLAY(DIDS.VID1,DIDS.PID,2,2)
      IF(.NOT.ISTAT)THEN
        ERRMSG='SMG$PASTE_VIRTUAL_DISPLAY (VID1)'
        GOTO 9000
      ENDIF
      ISTAT=SMG$PASTE_VIRTUAL_DISPLAY(DIDS.VID2,DIDS.PID,4,2)
      IF(.NOT.ISTAT)THEN
        ERRMSG='SMG$PASTE_VIRTUAL_DISPLAY (VID2)'
        GOTO 9000
      ENDIF

C
C ------display initial info-------------------------------------------
C
      ISTAT=SMG$PUT_CHARS(DIDS.VID1,'Seismic Data Base
      1 Information Pgm V6.1',1,20)
      IF(.NOT.ISTAT)THEN
        ERRMSG='SMG$PUT_CHARS <1>'
        GOTO 9000
      ENDIF

C
C ------set output direction-------------------------------------------
C
      CALL USRFILES_FLOW(USRINFO.FLOW)
      IF(USRINFO.FLOW.EQ.'EXIT')EOPGM=.TRUE.
C
C ---------------------------------------------------------------------
C     USER INPUT
C ---------------------------------------------------------------------
C
      DO WHILE(.NOT.EOPGM)

C
```

```
C --------display info-----------------------------------------------------
C
       DO WHILE((.NOT.EOPGM).AND.(.NOT.PROJECTLG))
         ISTAT=SMG$ERASE_DISPLAY(DIDS.VID2,1,1)
         ISTAT=SMG$PUT_CHARS(DIDS.VID2,'All default responses are shown in
     1 brackets: [DEFAULT].',2,1)
         IF(.NOT.ISTAT)THEN
           ERRMSG='SMG$PUT_CHARS (VID2) <2>'
           GOTO 9000
         ENDIF
         ISTAT=SMG$PUT_CHARS(DIDS.VID2,'Default response taken by hitting
     1 <CARRIAGE RETURN>.',3,1)
         IF(.NOT.ISTAT)THEN
           ERRMSG='SMG$PUT_CHARS (VID2) <3>'
           GOTO 9000
         ENDIF
         ISTAT=SMG$PUT_CHARS(DIDS.VID2,'To stop program execution, enter
     1 EXIT to any prompt.',4,1)
         IF(.NOT.ISTAT)THEN
           ERRMSG='SMG$PUT_CHARS (VID2) <4>'
           GOTO 9000
         ENDIF
         ISTAT=SMG$DRAW_LINE(DIDS.VID2,6,1,6,78)
         IF(.NOT.ISTAT)THEN
           ERRMSG='SMG$PUT_CHARS (VID2) <5>'
           GOTO 9000
         ENDIF
C
         ISTAT=SMG$END_DISPLAY_UPDATE(DIDS.VID2)
         IF(.NOT.ISTAT)THEN
           ERRMSG='SMG$END_DISPLAY_UPDATE (VID2)'
           GOTO 9000
         ENDIF

C
C --------enter project-----------------------------------------------------
C
         IROW=7
         ICOL=1
         ISTAT=SMG$ERASE_DISPLAY(DIDS.VID2,IROW,ICOL)
         IF(.NOT.ISTAT)THEN
           ERRMSG='SMG$ERASE_DISPLAY (VID2)'
           GOTO 9000
         ENDIF
C
         STR='enter <SEISDATA PROJECT>, exit.....['//USRINFO.PROJECT//']:'
         ISTAT=SMG$PUT_CHARS(DIDS.VID2,STR(1:46),8,1)
         ISTAT=SMG$RING_BELL(DIDS.VID2)
         ISTAT=SMG$READ_STRING(DIDS.KBID,PROJECT,,9)
         IF(PROJECT.EQ.' ')THEN
           PROJECT=USRINFO.PROJECT
         ELSE
           USRINFO.PROJECT=PROJECT
         END IF
```

```
          IF(PROJECT.EQ.'EXIT')THEN
            EOPGM=.TRUE.
          ELSE
            CALL SDB_SDBOPN(PROJECT,ISTAT)
            IF((ISTAT.EQ.SDB_NORMAL).OR.(ISTAT.EQ.SDB_READ))THEN
              PROJECTLG=.TRUE.
            ELSE
              CALL SDB_SDBMSG(ISTAT,TEXT)
         ISTAT=SMG$ERASE_DISPLAY(DIDS.VID2,1,1,20,78)
              IF(.NOT.ISTAT)CALL LIB$STOP(%VAL(ISTAT))
C
         ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(1:50),2,1,1)
         STR='hit <CARRIAGE RETRUN> to continue'
         ISTAT=SMG$PUT_CHARS(DIDS.VID2,STR(1:32),4,1)
         ISTAT=SMG$RING_BELL(DIDS.VID2,20)
         ISTAT=SMG$READ_STRING(DIDS.KBID,DUMMY)
            ENDIF
          ENDIF
        ENDDO

C
C --------data type entry-----------------------------------------
C
        IF(.NOT.EOPGM)THEN
          DO WHILE((.NOT.EOPGM).AND.(.NOT.DATYPELG))
            ISTAT=SMG$ERASE_DISPLAY(DIDS.VID2,1,1,20,78)
            ISTAT=SMG$PUT_CHARS(DIDS.VID2,'Project: '//PROJECT,2,1)
            ISTAT=SMG$PUT_CHARS(DIDS.VID2,'FILE TYPE:                ENTER:'
     1,4,1)
            ISTAT=SMG$PUT_CHARS(DIDS.VID2,'-------------             ------'
     1,5,1)
            ISTAT=SMG$PUT_CHARS(DIDS.VID2,'data set.................SET'
     1,6,1)
            ISTAT=SMG$PUT_CHARS(DIDS.VID2,'velocity file............VEL'
     1,7,1)
            ISTAT=SMG$PUT_CHARS(DIDS.VID2,'geometry file............GEOM'
     1,8,1)
            ISTAT=SMG$PUT_CHARS(DIDS.VID2,'filter file..............FLT'
     1,9,1)
            STR='enter SET, VEL, GEOM, exit.....['//USRINFO.DATYPE//']:'
            ISTAT=SMG$PUT_CHARS(DIDS.VID2,STR(1:38),11,1)
            ISTAT=SMG$RING_BELL(DIDS.VID2)
            ISTAT=SMG$READ_STRING(DIDS.KBID,DATYPE)
C
            IF(DATYPE.EQ.' ')THEN
          DATYPE=USRINFO.DATYPE
            ELSE
          USRINFO.DATYPE=DATYPE
            END IF
C
            IF(DATYPE.EQ.'EXIT')EOPGM=.TRUE.
            IF((DATYPE.EQ.'SET').OR.(DATYPE.EQ.'VEL').OR.
     1(DATYPE.EQ.'GEOM').OR.(DATYPE.EQ.'FLT'))DATYPELG=.TRUE.
          ENDDO
```

```
          IF(DATYPE.EQ.'GEOM')THEN
            LINEID='GEOMETRY'
               USRINFO.LINEID=LINEID
          ENDIF
        ENDIF

C
C --------line id-----------------------------------------------
C
        IF((.NOT.EOPGM).AND.(DATYPE.NE.'GEOM'))THEN
          ISTAT=SMG$ERASE_DISPLAY(DIDS.VID2,1,1,20,78)
          ISTAT=SMG$PUT_CHARS(DIDS.VID2,'Project: '//PROJECT,2,1,1)
          ISTAT=SMG$PUT_CHARS(DIDS.VID2,'Data type: '//DATYPE,2,30)
          STR='enter <LINE ID>, *, exit.....['//USRINFO.LINEID//']:'
          ISTAT=SMG$PUT_CHARS(DIDS.VID2,STR(1:41),6,1)
          ISTAT=SMG$RING_BELL(DIDS.VID2)
          ISTAT=SMG$READ_STRING(DIDS.KBID,LINEID,,8)
C
          IF(LINEID.EQ.' ')THEN
            LINEID=USRINFO.LINEID
          ELSE
            USRINFO.LINEID=LINEID
          END IF
C
          IF(LINEID.EQ.'EXIT')EOPGM=.TRUE.
        ENDIF

C
C --------execute pgm option selected---------------------------
C
        IF(.NOT.EOPGM)THEN
          IF(DATYPE.EQ.'SET')THEN
            CALL USRFILES_SET(EOPGM)
          ELSEIF(DATYPE.EQ.'VEL')THEN
            CALL USRFILES_VEL(EOPGM)
          ELSEIF(DATYPE.EQ.'FLT')THEN
            CALL USRFILES_FLT(EOPGM)
          ELSE
            CALL USRFILES_GEOM(EOPGM)
          ENDIF
C
          IF(.NOT.EOPGM)THEN
            PROJECTLG=.FALSE.
            DATYPELG=PROJECTLG
          ENDIF
        ENDIF
C
      ENDDO   ! END MAIN WHILE LOOP
      GOTO 9999
C
9000  WRITE(6,909)ERRMSG,ISTAT
      CALL LIB$STOP(%VAL(ISTAT))
C
C ------------------------------------------------------------
```

```
C
9999 INQUIRE(13,OPENED=OPLOG)
     IF(OPLOG)THEN
       CLOSE(13,STATUS='PRINT')
       ISTAT=SMG$ERASE_DISPLAY(DIDS.VID2,'output file:
     1 SEISDATA_INFO.LIS',2,1)
     ENDIF
     STOP 'YO MAMA'
C
909  FORMAT(' run-time routine ', A,' returned a value of ', Z8)
     END

C
C ================================================================
C ================================================================
C    SUBROUTINE USRFILES_FLOW: subr called from USRFILES_MAIN
C    written by F.N. Zihlman, Feb. 1986
C ----------------------------------------------------------------
C    Modified to use the V4 series Screen Management runtime
C    libraries and to use STRUCTURES and RECORDS.
C    F.N. Zihlman APR. 1986
C ----------------------------------------------------------------
C
C    FUNCTION: select output direction for data to SCREEN, FILE or
C         BOTH.
C ----------------------------------------------------------------
C
     SUBROUTINE USRFILES_FLOW(DFLOW)
C
     STRUCTURE /DINFO/
          INTEGER PID,KBID,VID1,VID2
     END STRUCTURE
C
     RECORD /DINFO/DIDS
C
     CHARACTER*6 FLOW,DFLOW
C
     INTEGER*4 SMG$ERASE_DISPLAY
     INTEGER*4 SMG$PUT_CHARS
     INTEGER*4 SMG$RING_BELL
     INTEGER*4 SMG$READ_STRING
C
     LOGICAL EOPROC
C
     COMMON /DFO/DIDS
C
C ----------------------------------------------------------------
C
     EOPROC=.FALSE.
     DO WHILE(.NOT.EOPROC)
       ISTAT=SMG$ERASE_DISPLAY(DIDS.VID2,1,1)
       ISTAT=SMG$PUT_CHARS(DIDS.VID2,'Seisdata data base information may
     1 be displayed to either the terminal screen,',2,1)
       ISTAT=SMG$PUT_CHARS(DIDS.VID2,'a file on disk which may be saved
```

```
     1 and/or printed, or both.',3,1)
       ISTAT=SMG$PUT_CHARS(DIDS.VID2,'enter SCREEN, FILE, or BOTH
     1.....['//DFLOW//']: ',5,1)
       ISTAT=SMG$RING_BELL(DIDS.KBID,1)
       ISTAT=SMG$READ_STRING(DIDS.KBID,FLOW)
C
       IF(FLOW.EQ.' ')THEN
         FLOW=DFLOW
       ELSE
         DFLOW=FLOW
       ENDIF
C
       IF((FLOW.EQ.'SCREEN').OR.(FLOW.EQ.'FILE').OR.(FLOW.EQ.'BOTH')
     1.OR.(FLOW.EQ.'EXIT'))THEN
           IF((FLOW.EQ.'FILE').OR.(FLOW.EQ.'BOTH'))THEN
             OPEN(13,FILE='SEISDATA_INFO.LIS',CARRIAGECONTROL='LIST'
     1,STATUS='NEW')
           ENDIF
           EOPROC=.TRUE.
       ENDIF
     ENDDO
     RETURN
     END

C
C ================================================================
C ================================================================
C    SUBROUTINE USRFILES_SET: subr called from USRFILES_MAIN
C    written by F.N. Zihlman, Feb. 1986
C ----------------------------------------------------------------
C    Modified to use the V4 series Screen Management runtime
C    libraries and to use STRUCTURES and RECORDS.
C    F.N. Zihlman APR. 1986
C ----------------------------------------------------------------
C
C    FUNCTION: select all TAPOUT class entries matching the given LINEID,
C         TAPOnn, and FILEKEY values, open the returned FILENAME and
C         display the desired information. The selection and display
C         process is continued until no more matches are found.
C
     SUBROUTINE USRFILES_SET(EOPGM)
C
     LOGICAL EOPGM,KEYIDLG,VERNOLG
C
     STRUCTURE /UINFO/
          CHARACTER*56 KEYID
          CHARACTER*8  PROJECT,LINEID
          CHARACTER*6  FLOW
          CHARACTER*4  VERNO,DATYPE
     END STRUCTURE
C
```

```
      STRUCTURE /SINFO/
           CHARACTER*80 DESCRIP,FILENAME
           CHARACTER*56 KEY
           CHARACTER*24 SDDATE
           CHARACTER*8  LINE,CLASS
      END STRUCTURE
C
      STRUCTURE /DINFO/
           INTEGER PID,KBID,VID1,VID2
      END STRUCTURE
C
      CHARACTER*80 TEXT
      CHARACTER*56 KEYID
      CHARACTER*8  PROJECT,LINEID,CLASSID
      CHARACTER*4  VERNO
      CHARACTER*1  DUMMY
C
      INTEGER*4 SMG$ERASE_DISPLAY
      INTEGER*4 SMG$PUT_CHARS
      INTEGER*4 SMG$RING_BELL
      INTEGER*4 SMG$READ_STRING
      RECORD /UINFO/USRINFO
      RECORD /SINFO/SEISINFO
      RECORD /DINFO/DIDS
C
      COMMON /UFO/USRINFO
      COMMON /SFO/SEISINFO
      COMMON /DFO/DIDS
C
C ---------------------------------------------------------------------
C
      KEYIDLG=.FALSE.
      VERNOLG=.FALSE.
C
      DO WHILE((.NOT.EOPGM).AND.(.NOT.KEYIDLG))
        ISTAT=SMG$ERASE_DISPLAY(DIDS.VID2,1,1)
        TEXT='project: '//USRINFO.PROJECT//'         line: '
     1//USRINFO.LINEID
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(1:38),2,1)
        TEXT='data type selected: SET'
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(1:24),4,1)
C
        TEXT='enter <DATA SET NAME>, *, exit...['//USRINFO.KEYID(1:16)
     1//']: '
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(1:53),6,1)
        ISTAT=SMG$RING_BELL(DIDS.VID2,1)
        ISTAT=SMG$READ_STRING(DIDS.KBID,KEYID,,17)
C
        IF(KEYID.EQ.' ')THEN
          KEYID=USRINFO.KEYID
        ELSE
          USRINFO.KEYID=KEYID
        ENDIF
C
```

```
        IF(KEYID.EQ.'EXIT')THEN
          EOPGM=.TRUE.
        ELSE
          KEYIDLG=.TRUE.
        ENDIF
      ENDDO
C
C
      DO WHILE((.NOT.EOPGM).AND.(.NOT.VERNOLG))
        ISTAT=SMG$ERASE_DISPLAY(DIDS.VID2,1,1)
        TEXT='project: '//USRINFO.PROJECT//'          line: '
     1//USRINFO.LINEID
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(1:38),2,1)
        TEXT='data type selected: SET                set name: '
     1//USRINFO.KEYID(1:16)
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(1:78),4,1)
        TEXT='enter <VERSION NUMBER>, *, exit.....['
     1//USRINFO.VERNO//']: '
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(1:44),6,1)
        ISTAT=SMG$RING_BELL(DIDS.VID2,1)
        ISTAT=SMG$READ_STRING(DIDS.KBID,VERNO,,5)
C
        IF(VERNO.EQ.' ')THEN
          VERNO=USRINFO.VERNO
        ELSE
          USRINFO.VERNO=VERNO
        ENDIF
C
        IF(VERNO.EQ.'EXIT')THEN
          EOPGM=.TRUE.
        ELSE
          IDX=INDEX(VERNO,' ')-1
          IF(IDX.EQ.1)THEN
            CLASSID='TAPOO'//VERNO(1:1)
            VERNOLG=.TRUE.
          ELSEIF(IDX.EQ.2)THEN
            CLASSID='TAPO'//VERNO(1:2)
            VERNOLG=.TRUE.
          ENDIF
        ENDIF
      ENDDO
C
C
      IF(.NOT.EOPGM)THEN
        CALL SDB_FILMSK(USRINFO.LINEID,CLASSID,USRINFO.KEYID,ICTX,ISTAT)
        IF(ISTAT.NE.SDB_NORMAL)THEN
          CALL SDB_SDBMSG(ISTAT,TEXT)
          ISTAT=SMG$ERASE_DISPLAY(DIDS.VID2,1,1)
          ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(1:78),2,1)
          TEXT='hit <CARRIAGE RETURN> to continue'
          ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(1:40),4,1)
```

```
          ISTAT=SMG$READ_STRING(DIDS.KBID,DUMMY,,1)
          RETURN
        ENDIF
      ENDIF
C
C
      DO WHILE(ISTAT.EQ.SDB_NORMAL)
        CALL SDB_FILNXT(ICTX,SEISINFO.LINE,SEISINFO.CLASS
     1,SEISINFO.KEY,SEISINFO.DESCRIP,SEISINFO.FILENAME
     2,SEISINFO.SDDATE,ISTAT)
        IF(ISTAT.EQ.SDB_NORMAL)THEN
          CALL USRFILES_SETDSP(USRINFO.FLOW)
        ELSEIF(ISTAT.NE.SDB_NMF)THEN
          CALL SDB_SDBMSG(ISTAT,TEXT)
          ISTAT=SMG$ERASE_DISPLAY(DIDS.VID2,1,1)
          ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(1:78),2,1)
          TEXT='hit <CARRIAGE RETURN> to continue'
          ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(1:40),4,1)
          ISTAT=SMG$READ_STRING(DIDS.KBID,DUMMY,,1)
        ENDIF
      ENDDO
      RETURN
C
C ---------------------------------------------------------------
C
      END

C
C ===============================================================
C ===============================================================
C     USRFILES_SETDSP: called from USRFILES_SET subroutine
C
C     FUNCTION: to display all TAPOnn files found in USRFILES_SET showing
C          the desired information
C
C
      SUBROUTINE USRFILES_SETDSP(FLOW)
C
      STRUCTURE /SINFO/
         CHARACTER*80 DESCRIP,FILENAME
         CHARACTER*56 KEY
         CHARACTER*24 SDDATE
         CHARACTER*8  LINE,CLASS
      END STRUCTURE
C
      STRUCTURE /DINFO/
         INTEGER*4 PID
         INTEGER*4 KBID
         INTEGER*4 VID1
         INTEGER*4 VID2
      END STRUCTURE
C
      RECORD /SINFO/SEISINFO
      RECORD /DINFO/DIDS
```

```
C
      CHARACTER*78 STR,TEXT(5),ERRMSG
      CHARACTER*4  DUMMY
      CHARACTER*65 MSG
      CHARACTER*80 RECRD,SEPSTR
      CHARACTER*8 DNSTY,SORDER,REEL1,HDRCNT,CMPHDR,SAMPLES,SAMPRATE
      1,DATALEN
      CHARACTER*6 EN1ST,EN2ND,TRACES,FLOW
      CHARACTER*3 TR1,TR2,CCOUNT
C
      INTEGER*4 SMG$PUT_CHARS
      INTEGER*4 SMG$BEGIN_DISPLAY_UPDATE,SMG$END_DISPLAY_UPDATE
      INTEGER*4 SMG$ERASE_DISPLAY
      INTEGER*4 SMG$READ_STRING
C
      COMMON /SFO/SEISINFO
      COMMON /DFO/DIDS
C
C ---------------------------------------------------------------------
C
      SEPSTR='--------------------------------------------------
      1----------------------------'
C
      TEXT(1)='line: '//SEISINFO.LINE//'     version: '//SEISINFO.CLASS(5:6)
      1//'     set: '//SEISINFO.KEY(1:16)
      TEXT(2)='description:'
      TEXT(3)=SEISINFO.DESCRIP
      TEXT(4)=SEISINFO.FILENAME
      TEXT(5)='date: '//SEISINFO.SDDATE
      IF((FLOW.EQ.'SCREEN').OR.(FLOW.EQ.'BOTH'))THEN
        ISTAT=SMG$ERASE_DISPLAY(DIDS.VID2,1,1)
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(1),1,1)
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(2),3,1)
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(3),4,1)
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(4),6,1)
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(5),7,1)
      ENDIF
      IF((FLOW.EQ.'FILE').OR.(FLOW.EQ.'BOTH'))THEN
        WRITE(13,'(A1)')' '
        WRITE(13,'(A80)')SEPSTR
        WRITE(13,'(A80)')SEPSTR
        WRITE(13,'(A1)')' '
        WRITE(13,'(A78)')TEXT(1)
        WRITE(13,'(A1)')' '
        WRITE(13,'(A78)')TEXT(2)
        WRITE(13,'(A78)')TEXT(3)
        WRITE(13,'(A1)')' '
        WRITE(13,'(A78)')TEXT(4)
        WRITE(13,'(A78)')TEXT(5)
        WRITE(13,'(A1)')' '
      ENDIF
C
C ---------------------------------------------------------------------
C
```

```
      ICOUNT=0
      IROW=13
      OPEN(12,FILE=SEISINFO.FILENAME,STATUS='OLD',READONLY,ERR=30,
     1IOSTAT=IOS)
C
10    READ(12,'(A80)',ERR=40,IOSTAT=IOS,END=20)RECRD
C
      IF(RECRD(1:8).EQ.'FORMAT')THEN
        DNSTY=RECRD(9:16)
      ELSE IF(RECRD(1:8).EQ.'DATA1')THEN
        SAMPLES=RECRD(33:40)
        SAMPRATE=RECRD(41:48)
        HDRCNT=RECRD(65:72)
C
        DECODE(8,107,SAMPLES,IOSTAT=IOS,ERR=50)ISAMPLES
        DECODE(8,107,SAMPRATE,IOSTAT=IOS,ERR=50)ISAMPRATE
        IDATALEN=ISAMPLES*ISAMPRATE/1000
        RSAMPRATE=ISAMPRATE/1000.
        ENCODE(8,'(I8)',DATALEN,IOSTAT=IOS,ERR=50)IDATALEN
        ENCODE(8,'(F8.3)',SAMPRATE,IOSTAT=IOS,ERR=50)RSAMPRATE
C
      ELSE IF(RECRD(1:8).EQ.'HDR')THEN
        CMPHDR=RECRD(17:24)
        IF(CMPHDR.EQ.HDRCNT)THEN
          SORDER=RECRD(9:16)
C ---------------------------------------------------------------------
          TEXT(1)='density: '//DNSTY//'   sort order: '//SORDER
          TEXT(2)='sample rate (MS): '//SAMPRATE(3:8)//'     samples: '
     1//SAMPLES(3:8)//'     data length (MS): '//DATALEN(3:8)
          TEXT(3)='                  first                    last
     1                  total'
          TEXT(4)='     reel         ensemble/trace           ensemble/trace
     1          traces'
          IF((FLOW.EQ.'SCREEN').OR.(FLOW.EQ.'BOTH'))THEN
            ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(1),9,1)
            ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(2),10,1)
            ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(3),12,1)
            ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(4),13,1)
          ENDIF
          IF((FLOW.EQ.'FILE').OR.(FLOW.EQ.'BOTH'))THEN
            WRITE(13,'(A80)')TEXT(1)
            WRITE(13,'(A80)')TEXT(2)
            WRITE(13,'(A1)')' '
            WRITE(13,'(A80)')TEXT(3)
            WRITE(13,'(A80)')TEXT(4)
          END IF
        END IF
      ELSE IF(RECRD(1:8).EQ.'REEL')THEN
        REEL1=RECRD(25:32)
```

```
      ELSE IF(RECRD(1:3).EQ.' ')THEN
        EN1ST=RECRD(7:12)
        TR1=RECRD(23:25)
        EN2ND=RECRD(36:41)
        TR2=RECRD(52:54)
        TRACES=RECRD(55:60)
C
        ICOUNT=ICOUNT+1
        IROW=IROW+1
        TEXT(1)=REEL1//'       '//EN1ST//'   '//TR1//'            '
     1//EN2ND//'   '//TR2//'          '//TRACES
        IF(ICOUNT.LT.10)THEN
          IBYTES=1
        ELSEIF((ICOUNT.GE.10).AND.(ICOUNT.LT.100))THEN
          IBYTES=2
        ELSE
          IBYTES=3
        ENDIF
        ENCODE(IBYTES,'(I<IBYTES>)',CCOUNT)ICOUNT
        IF(ICOUNT.LT.10)THEN
          TEXT(1)=' '//CCOUNT(1:1)//')  '//TEXT(1)
        ELSE
          TEXT(1)=CCOUNT(1:2)//')  '//TEXT(1)
        ENDIF
        IF((FLOW.EQ.'SCREEN').OR.(FLOW.EQ.'BOTH'))THEN
          ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(1),IROW,1)
          IF(IROW.EQ.18)THEN
            ISTAT=SMG$PUT_CHARS(DIDS.VID2,'hit <CARRIAGE RETURN> to
     1 continue',20,1)
            ISTAT=SMG$READ_STRING(DIDS.KBID,DUMMY)
            ISTAT=SMG$ERASE_DISPLAY(DIDS.VID2,11,1,20,78)
            IROW=12
          ENDIF
        ENDIF
        IF((FLOW.EQ.'FILE').OR.(FLOW.EQ.'BOTH'))THEN
          WRITE(13,'(A80)')TEXT(1)
        ENDIF
      END IF
C
      GOTO 10
C ---------------------------------------------------------------
C
20    CLOSE(12)
      IF((FLOW.EQ.'SCREEN').OR.(FLOW.EQ.'BOTH'))THEN
        ISTAT=SMG$PUT_CHARS(DIDS.VID2
     1,'hit <CARRIAGE RETURN> to continue',20,1)
        ISTAT=SMG$READ_STRING(DIDS.KBID,DUMMY)
      ENDIF
      RETURN
C
C ---------------------------------------------------------------
C
30    MSG=' OPEN ERROR USING '//SEISINFO.FILENAME
      GOTO 60
```

```
C
40    MSG=' READ ERROR USING '//SEISINFO.FILENAME
      CLOSE(12)
      GOTO 60
C
50    MSG=' ERROR IN DECODE/ENCODE STATEMENT'
      CLOSE(12)
C
60    IF((FLOW.EQ.'SCREEN').OR.(FLOW.EQ.'BOTH'))THEN
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,MSG,19,1)
      ENDIF
      IF((FLOW.EQ.'PRINT').OR.(FLOW.EQ.'BOTH'))THEN
        WRITE(13,104) MSG
        WRITE(13,106) IOS
      END IF
C
C
      IF((FLOW.EQ.'SCREEN').OR.(FLOW.EQ.'BOTH'))THEN
        ISTAT=SMG$PUT_CHARS(DIDS.VID2
      1,'hit <CARRIAGE RETURN> to continue',20,1)
        ISTAT=SMG$READ_STRING(DIDS.KBID,DUMMY)
      ENDIF
      RETURN
C
104   FORMAT(A65)
106   FORMAT(' ERROR NUMBER ',I3)
107   FORMAT(I8)
C
      END

C
C ================================================================
C ================================================================
C     SUBROUTINE USRFILES_VEL: subr called from USRFILES_MAIN
C     written by F.N. Zihlman, Feb. 1986
C ----------------------------------------------------------------
C     Modified to use the V4 series Screen Management runtime
C     libraries and to use STRUCTURES and RECORDS.
C     F.N. Zihlman APR. 1986
C ----------------------------------------------------------------
C
C     FUNCTION: select all VELDEFN class entries matching the given LINEID,
C          and VELOCITY ID values, open the returned FILENAME and
C          display the desired information. The selection and display
C          process is continued until no more matches are found.
C
      SUBROUTINE USRFILES_VEL(EOPGM)
C
      LOGICAL EOPGM,KEYIDLG
C
```

```
      STRUCTURE /UINFO/
           CHARACTER*56 KEYID
           CHARACTER*8  PROJECT,LINEID
           CHARACTER*6  FLOW
           CHARACTER*4  VERNO,DATYPE
      END STRUCTURE
C
      STRUCTURE /SINFO/
           CHARACTER*80 DESCRIP,FILENAME
           CHARACTER*56 KEY
           CHARACTER*24 SDDATE
           CHARACTER*8  LINE,CLASS
      END STRUCTURE
C
      STRUCTURE /DINFO/
           INTEGER PID,KBID,VID1,VID2
      END STRUCTURE
C
      CHARACTER*80 TEXT
      CHARACTER*56 KEYID
      CHARACTER*8  PROJECT,LINEID,CLASSID
      CHARACTER*1  DUMMY
C
      INTEGER*4 SMG$ERASE_DISPLAY
      INTEGER*4 SMG$PUT_CHARS
      INTEGER*4 SMG$RING_BELL
      INTEGER*4 SMG$READ_STRING
      RECORD /UINFO/USRINFO
      RECORD /SINFO/SEISINFO
      RECORD /DINFO/DIDS
C
      COMMON /UFO/USRINFO
      COMMON /SFO/SEISINFO
      COMMON /DFO/DIDS
C
C ---------------------------------------------------------------------
C
      KEYIDLG=.FALSE.
C
C
      ISTAT=SMG$ERASE_DISPLAY(DIDS.VID2,1,1)
      TEXT='project: '//USRINFO.PROJECT//'          line: '
      1//USRINFO.LINEID
      ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(1:42),2,1)
      TEXT='data type selected: VELDEFN'
      ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(1:42),4,1)
C
      DO WHILE((.NOT.EOPGM).AND.(.NOT.KEYIDLG))
        TEXT='enter <VELOCITY ID>, *, exit.....['//USRINFO.KEYID(1:8)//']: '
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(1:46),6,1)
        ISTAT=SMG$RING_BELL(DIDS.VID2,1)
        ISTAT=SMG$READ_STRING(DIDS.KBID,KEYID,,9)
C
```

```
        IF(KEYID.EQ.' ')THEN
          KEYID=USRINFO.KEYID
        ELSE
          USRINFO.KEYID=KEYID
        ENDIF
C
        IF(KEYID.EQ.'EXIT')THEN
          EOPGM=.TRUE.
        ELSE
          KEYIDLG=.TRUE.
        ENDIF
      ENDDO
C
C
      IF(.NOT.EOPGM)CLASSID='VELDEFN'
C
C
      IF(.NOT.EOPGM)THEN
        CALL SDB_FILMSK(USRINFO.LINEID,CLASSID,USRINFO.KEYID,ICTX,ISTAT)
        IF(ISTAT.NE.SDB_NORMAL)THEN
          CALL SDB_SDBMSG(ISTAT,TEXT)
          ISTAT=SMG$ERASE_DISPLAY(DIDS.VID2,19,1)
          ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT,19,1)
          ISTAT=SMG$PUT_CHARS(DIDS.VID2,'hit <CARRIAGE RETURN>
     1 to continue',20,1)
          ISTAT=SMG$READ_STRING(DIDS.KBID,DUMMY)
          RETURN
        ENDIF
      ENDIF
C
C
      DO WHILE(ISTAT.EQ.SDB_NORMAL)
        CALL SDB_FILNXT(ICTX,SEISINFO.LINE,SEISINFO.CLASS
     1,SEISINFO.KEY,SEISINFO.DESCRIP,SEISINFO.FILENAME
     2,SEISINFO.SDDATE,ISTAT)
C
        IF(ISTAT.EQ.SDB_NORMAL)THEN
          CALL USRFILES_VELDSP(USRINFO.FLOW)
        ELSEIF(ISTAT.NE.SDB_NMF)THEN
          CALL SDB_SDBMSG(ISTAT,TEXT)
          ISTAT=SMG$ERASE_DISPLAY(DIDS.VID2,19,1)
          ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT,19,1)
          ISTAT=SMG$PUT_CHARS(DIDS.VID2,'hit <CARRIAGE RETURN>
     1 to continue',20,1)
          ISTAT=SMG$READ_STRING(DIDS.KBID,DUMMY)
        ENDIF
      ENDDO
      RETURN
C
      END

C
```

```
C ================================================================
C ================================================================
C    USRFILES_VELDSP: called from USRFILES_VEL subroutine
C
C    FUNCTION: to display all VELDEFN files found in USRFILES_VEL showing
C         the desired information
C
C
      SUBROUTINE USRFILES_VELDSP(FLOW)
C
      STRUCTURE /SINFO/
         CHARACTER*80 DESCRIP,FILENAME
         CHARACTER*56 KEY
         CHARACTER*24 SDDATE
         CHARACTER*8  LINE,CLASS
      END STRUCTURE
C
      STRUCTURE /DINFO/
         INTEGER*4 PID
         INTEGER*4 KBID
         INTEGER*4 VID1
         INTEGER*4 VID2
      END STRUCTURE
C
      RECORD /SINFO/SEISINFO
      RECORD /DINFO/DIDS
C
      INTEGER*4 SMG$PUT_CHARS
      INTEGER*4 SMG$ERASE_DISPLAY
      INTEGER*4 SMG$READ_STRING
C
      CHARACTER*4  DUMMY
      CHARACTER*6  FLOW
      CHARACTER*78 TEXT(5),SEPSTR
C
      COMMON /SFO/SEISINFO
      COMMON /DFO/DIDS
C
C ---------------------------------------------------------------
C
      SEPSTR='--------------------------------------------------
     1----------------------------'
C
      TEXT(1)='line: '//SEISINFO.LINE//'     version: '//SEISINFO.CLASS
     1//'     set: '//SEISINFO.KEY(1:16)
      TEXT(2)='description:'
      TEXT(3)=SEISINFO.DESCRIP
      TEXT(4)=SEISINFO.FILENAME
      TEXT(5)='date: '//SEISINFO.SDDATE
C
      IF((FLOW.EQ.'SCREEN').OR.(FLOW.EQ.'BOTH'))THEN
        ISTAT=SMG$ERASE_DISPLAY(DIDS.VID2,1,1)
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(1),1,1)
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(2),3,1)
```

```
        ISTAT=SMG$PUT_CHARS(DIDS.VID2.TEXT(3),4,1)
        ISTAT=SMG$PUT_CHARS(DIDS.V1D2,TEXT(4),6,1)
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(5),7,1)
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,'hit carriage return to
     1 continue',9,1)
        ISTAT=SMG$READ_STRING(DIDS.KBID,DUMMY)
      ENDIF
      IF((FLOW.EQ.'FILE').OR.(FLOW.EQ.'BOTH'))THEN
        WRITE(13,'(A80)')SEPSTR
        WRITE(13,'(A80)')SEPSTR
        WRITE(13,'(A1)')
        WRITE(13,'(A80)')TEXT(1)
        WRITE(13,'(A1)')
        WRITE(13,'(A80)')TEXT(2)
        WRITE(13,'(A80)')TEXT(3)
        WRITE(13,'(A1)')
        WRITE(13,'(A80)')TEXT(4)
        WRITE(13,'(A80)')TEXT(5)
        WRITE(13,'(A80)')
      ENDIF
C
      RETURN
      END

C
C ================================================================
C ================================================================
C     SUBROUTINE USRFILES_FLT: subr called from USRFILES_MAIN
C     written by F.N. Zihlman, Feb. 1986
C ----------------------------------------------------------------
C     Modified to use the V4 series Screen Management runtime
C     libraries and to use STRUCTURES and RECORDS.
C     F.N. Zihlman APR. 1986
C ----------------------------------------------------------------
C
C     FUNCTION: select all FILTER22 class entries matching the given LINEID,
C          and FILTER ID values, open the returned FILENAME and
C          display the desired information. The selection and display
C          process is continued until no more matches are found.
C
      SUBROUTINE USRFILES_FLT(EOPGM)
C
      LOGICAL EOPGM,KEYIDLG
C
      STRUCTURE /UINFO/
           CHARACTER*56 KEYID
           CHARACTER*8  PROJECT,LINEID
           CHARACTER*6  FLOW
           CHARACTER*4  VERNO,DATYPE
      END STRUCTURE
C
```

```
      STRUCTURE /SINFO/
           CHARACTER*80 DESCRIP,FILENAME
           CHARACTER*56 KEY
           CHARACTER*24 SDATE
           CHARACTER*8  LINE,CLASS
      END STRUCTURE
C
      STRUCTURE /DINFO/
           INTEGER PID,KBID,VID1,VID2
      END STRUCTURE
C
      CHARACTER*80 TEXT
      CHARACTER*56 KEYID
      CHARACTER*8  PROJECT,LINEID,CLASSID
      CHARACTER*1  DUMMY
C
      INTEGER*4 SMG$ERASE_DISPLAY
      INTEGER*4 SMG$PUT_CHARS
      INTEGER*4 SMG$RING_BELL
      INTEGER*4 SMG$READ_STRING
      RECORD /UINFO/USRINFO
      RECORD /SINFO/SEISINFO
      RECORD /DINFO/DIDS
C
      COMMON /UFO/USRINFO
      COMMON /SFO/SEISINFO
      COMMON /DFO/DIDS
C
C ---------------------------------------------------------------
C
      KEYIDLG=.FALSE.
C
C
      ISTAT=SMG$ERASE_DISPLAY(DIDS.VID2,1,1)
      TEXT='project: '//USRINFO.PROJECT//'          line: '
     1//USRINFO.LINEID
      ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(1:42),2,1)
      TEXT='data type selected: FILTER'
      ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(1:42),4,1)
C
      DO WHILE((.NOT.EOPGM).AND.(.NOT.KEYIDLG))
        TEXT='enter <FILTER ID>, *, exit.....['//USRINFO.KEYID(1:8)//']: '
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(1:46),6,1)
        ISTAT=SMG$RING_BELL(DIDS.VID2,1)
        ISTAT=SMG$READ_STRING(DIDS.KBID,KEYID)
C
        IF(KEYID.EQ.' ')THEN
          KEYID=USRINFO.KEYID
        ELSE
          USRINFO.KEYID=KEYID
        ENDIF
C
```

```
        IF(KEYID.EQ.'EXIT')THEN
          EOPGM=.TRUE.
        ELSE
          KEYIDLG=.TRUE.
        ENDIF
      ENDDO
C
C
      IF(.NOT.EOPGM)CLASSID='FILTER22'
C
C
      IF(.NOT.EOPGM)THEN
        CALL SDB_FILMSK(USRINFO.LINEID,CLASSID,USRINFO.KEYID,ICTX,ISTAT)
        IF(ISTAT.NE.SDB_NORMAL)THEN
          CALL SDB_SDBMSG(ISTAT,TEXT)
          ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT,19,1)
          ISTAT=SMG$PUT_CHARS(DIDS.VID2,'hit <CARRIAGE RETURN>
     1 to continue',20,1)
          ISTAT=SMG$READ_STRING(DIDS.KBID,DUMMY)
          RETURN
        ENDIF
      ENDIF
C
C
      DO WHILE(ISTAT.EQ.SDB_NORMAL)
        CALL SDB_FILNXT(ICTX,SEISINFO.LINE,SEISINFO.CLASS
     1,SEISINFO.KEY,SEISINFO.DESCRIP,SEISINFO.FILENAME
     2,SEISINFO.SDATE,ISTAT)
C
        IF(ISTAT.EQ.SDB_NORMAL)THEN
          CALL USRFILES_FLTDSP(USRINFO.FLOW)
        ELSEIF(ISTAT.NE.SDB_NMF)THEN
          CALL SDB_SDBMSG(ISTAT,TEXT)
          ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT,19,1)
          ISTAT=SMG$PUT_CHARS(DIDS.VID2,'hit <CARRIAGE RETURN>
     1 to continue',20,1)
          ISTAT=SMG$READ_STRING(DIDS.KBID,DUMMY)
        ENDIF
      ENDDO
      RETURN
C
      END

C
C ================================================================
C ================================================================
C     USRFILES_FLTDSP: called from USRFILES_FLT subroutine
C
C     FUNCTION: to display all FILTER22 files found in USRFILES_FLT showing
C          the desired information
C
C
      SUBROUTINE USRFILES_FLTDSP(FLOW)
C
```

```
      STRUCTURE /SINFO/
         CHARACTER*80 DESCRIP,FILENAME
         CHARACTER*56 KEY
         CHARACTER*24 SDDATE
         CHARACTER*8  LINE,CLASS
      END STRUCTURE
C
      STRUCTURE /DINFO/
         INTEGER*4 PID
         INTEGER*4 KBID
         INTEGER*4 VID1
         INTEGER*4 VID2
      END STRUCTURE
C
      RECORD /SINFO/SEISINFO
      RECORD /DINFO/DIDS
C
      INTEGER*4 SMG$PUT_CHARS
      INTEGER*4 SMG$ERASE_DISPLAY
      INTEGER*4 SMG$READ_STRING
C
      CHARACTER*4  DUMMY
      CHARACTER*6  FLOW
      CHARACTER*80 TEXT(5)
C
      COMMON /SFO/SEISINFO
      COMMON /DFO/DIDS
C
C ---------------------------------------------------------------
C
      TEXT(1)='line: '//SEISINFO.LINE//'     version: '//SEISINFO.CLASS
     1//'     set: '//SEISINFO.KEY(1:16)
      TEXT(2)='description:'
      TEXT(3)='date: '//SEISINFO.SDDATE
      TEXT(4)=SEISINFO.FILENAME
C
      IF((FLOW.EQ.'SCREEN').OR.(FLOW.EQ.'BOTH'))THEN
        ISTAT=SMG$ERASE_DISPLAY(DIDS.VID2,1,1)
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(1),1,1)
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(2),3,1)
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(3),4,1)
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(4),6,1)
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(5),7,1)
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,'hit <CARRIAGE RETURN> to
     1 continue',9,1)
        ISTAT=SMG$READ_STRING(DIDS.KBID,DUMMY)
      ENDIF
      IF((FLOW.EQ.'FILE').OR.(FLOW.EQ.'BOTH'))THEN
        WRITE(13,'(A1)')
        WRITE(13,'(A80)')TEXT(1)
        WRITE(13,'(A1)')
        WRITE(13,'(A80)')TEXT(2)
        WRITE(13,'(A80)')TEXT(3)
        WRITE(13,'(A1)')
```

```
        WRITE(13,'(A80)')TEXT(4)
        WRITE(13,'(A80)')TEXT(5)
        WRITE(13,'(A1)')
      ENDIF
C
      RETURN
      END

C
C ================================================================
C ================================================================
C     SUBROUTINE USRFILES_GEOM: subr called from USRFILES_MAIN
C     written by F.N. Zihlman, Feb. 1986
C ----------------------------------------------------------------
C     Modified to use the V4 series Screen Management runtime
C     libraries and to use STRUCTURES and RECORDS.
C     F.N. Zihlman JUNE 1986
C ----------------------------------------------------------------
C
C     FUNCTION: select all GEOM class entries matching the given PATTERN
C          ID values, open the returned FILENAME and
C          display the desired information. The selection and display
C          process is continued until no more matches are found.
C
C          LINDID="GEOMETRY"   CLASSID="PATTERN"   KEYID=<PATTERN NUMBER>
C
C ----------------------------------------------------------------
C ----------------------------------------------------------------
C
      SUBROUTINE USRFILES_GEOM(EOPGM)
C
      LOGICAL EOPGM,KEYIDLG
C
      STRUCTURE /UINFO/
           CHARACTER*56 KEYID
           CHARACTER*8  PROJECT,LINEID
           CHARACTER*6  FLOW
           CHARACTER*4  VERNO,DATYPE
      END STRUCTURE
C
      STRUCTURE /SINFO/
           CHARACTER*80 DESCRIP,FILENAME
           CHARACTER*56 KEY
           CHARACTER*24 SDATE
           CHARACTER*8  LINE,CLASS
      END STRUCTURE
C
      STRUCTURE /DINFO/
           INTEGER PID,KBID,VID1,VID2
      END STRUCTURE
C
```

```
      CHARACTER*80 TEXT
      CHARACTER*56 KEYID
      CHARACTER*8  PROJECT,LINEID,CLASSID
      CHARACTER*1  DUMMY
C
      INTEGER*4 SMG$ERASE_DISPLAY
      INTEGER*4 SMG$PUT_CHARS
      INTEGER*4 SMG$RING_BELL
      INTEGER*4 SMG$READ_STRING
      RECORD /UINFO/USRINFO
      RECORD /SINFO/SEISINFO
      RECORD /DINFO/DIDS
C
      COMMON /UFO/USRINFO
      COMMON /SFO/SEISINFO
      COMMON /DFO/DIDS
C
C ------------------------------------------------------------
C
      KEYIDLG=.FALSE.
      DO WHILE((.NOT.EOPGM).AND.(.NOT.KEYIDLG))
        ISTAT=SMG$ERASE_DISPLAY(DIDS.VID2,1,1)
        TEXT='project: '//USRINFO.PROJECT//'     line: '//USRINFO.LINEID
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT,2,1)
        TEXT='data type selected: GEOMETRY'
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT,4,1)
        TEXT='enter PATTERN NUMBER, *, exit.....['//USRINFO.KEYID(1:16)
      1//']: '
        ISTAT=SMG$RING_BELL(DIDS.VID2,1)
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(1:54),6,1)
        ISTAT=SMG$READ_STRING(DIDS.KBID,KEYID)
C
        IF(KEYID.EQ.' ')THEN
          KEYID=USRINFO.KEYID
        ELSE
          USRINFO.KEYID=KEYID
        ENDIF
C
        IF(KEYID.EQ.'EXIT')THEN
          EOPGM=.TRUE.
        ELSE
          KEYIDLG=.TRUE.
        ENDIF
      ENDDO
C
C
      IF(.NOT.EOPGM)CLASSID='PATTERN'
C
C
      IF(.NOT.EOPGM)THEN
        CALL SDB_FILMSK(USRINFO.LINEID,CLASSID,USRINFO.KEYID,ICTX,ISTAT)
        IF(ISTAT.NE.SDB_NORMAL)THEN
          CALL SDB_SDBMSG(ISTAT,TEXT)
          ISTAT=SMG$ERASE_DISPLAY(DIDS.VID2,1,1)
```

```
          ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT,2,1)
          TEXT='hit <CARRIAGE RETURN> to continue'
          ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT,4,1)
          ISTAT=SMG$READ_STRING(DIDS.KBID,DUMMY)
          RETURN
        ENDIF
      ENDIF
C
C
      DO WHILE(ISTAT.EQ.SDB_NORMAL)
        CALL SDB_FILNXT(ICTX,SEISINFO.LINE,SEISINFO.CLASS
     1,SEISINFO.KEY,SEISINFO.DESCRIP,SEISINFO.FILENAME
     2,SEISINFO.SDATE,ISTAT)
C
        IF(ISTAT.EQ.SDB_NORMAL)THEN
          CALL USRFILES_GEODSP(USRINFO.FLOW)
        ELSEIF(ISTAT.NE.SDB_NMF)THEN
          CALL SDB_SDBMSG(ISTAT,TEXT)
          ISTAT=SMG$ERASE_DISPLAY(DIDS.VID2,1,1)
          ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT,2,1)
          TEXT='hit <CARRIAGE RETURN> to continue'
          ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT,4,1)
          ISTAT=SMG$READ_STRING(DIDS.KBID,DUMMY)
        ENDIF
      ENDDO
      RETURN
      END

C
C ================================================================
C ================================================================
C     SUBROUTINE USRFILES_GEODSP: subr called from USRFILES_MAIN
C     written by F.N. Zihlman, Feb. 1986
C ----------------------------------------------------------------
C     Modified to use the V4 series Screen Management runtime
C     libraries and to use STRUCTURES and RECORDS.
C     F.N. Zihlman JUNE 1986
C ----------------------------------------------------------------
C
C     FUNCTION: display all GEOM class entries matching the given PATTERN
C          ID values, open the returned FILENAME.
C
C          LINDID="GEOMETRY"   CLASSID="PATTERN"   KEYID=<PATTERN NUMBER>
C
      SUBROUTINE USRFILES_GEODSP(FLOW)
C
      LOGICAL EOPGM,KEYIDLG
C
      STRUCTURE /SINFO/
          CHARACTER*80 DESCRIP,FILENAME
          CHARACTER*56 KEY
          CHARACTER*24 SDDATE
          CHARACTER*8  LINE,CLASS
      END STRUCTURE
```

```
C
      STRUCTURE /DINFO/
           INTEGER PID,KBID,VID1,VID2
      END STRUCTURE
C
      CHARACTER*78 TEXT(5),SEPSTR
      CHARACTER*56 KEYID
      CHARACTER*8  PROJECT,LINEID,CLASSID
      CHARACTER*6 FLOW
      CHARACTER*1  DUMMY
C
      INTEGER*4 SMG$ERASE_DISPLAY
      INTEGER*4 SMG$PUT_CHARS
      INTEGER*4 SMG$RING_BELL
      INTEGER*4 SMG$READ_STRING
C
      RECORD /SINFO/SEISINFO
      RECORD /DINFO/DIDS
C
      COMMON /SFO/SEISINFO
      COMMON /DFO/DIDS
C
C -------------------------------------------------------------
C
      SEPSTR='-------------------------------------------------
     1----------------------------'
C
      TEXT(1)=SEISINFO.LINE//'     '//SEISINFO.CLASS
     1//'     pattern: '//SEISINFO.KEY(1:16)
      TEXT(2)='description:'
      TEXT(3)=SEISINFO.DESCRIP
      TEXT(4)=SEISINFO.FILENAME
      TEXT(5)=SEISINFO.SDDATE
      IF((FLOW.EQ.'SCREEN').OR.(FLOW.EQ.'BOTH'))THEN
        ISTAT=SMG$ERASE_DISPLAY(DIDS.VID2,1,1)
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(1),1,1)
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(2),3,1)
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(3),4,1)
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(4),6,1)
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,TEXT(5),7,1)
        ISTAT=SMG$PUT_CHARS(DIDS.VID2,'hit <CARRIAGE RETURN> to continue'
     1,9,1)
        ISTAT=SMG$READ_STRING(DIDS.KBID,DUMMY)
      ENDIF
      IF((FLOW.EQ.'FILE').OR.(FLOW.EQ.'BOTH'))THEN
        WRITE(13,'(A80)')SEPSTR
        WRITE(13,'(A80)')SEPSTR
        WRITE(13,'(A1)')
        WRITE(13,'(A80)')TEXT(1)
        WRITE(13,'(A1)')
        WRITE(13,'(A80)')TEXT(2)
        WRITE(13,'(A80)')TEXT(3)
        WRITE(13,'(A1)')
        WRITE(13,'(A80)')TEXT(4)
```

```
  WRITE(13,'(A80)')TEXT(5)
  WRITE(13,'(A1)')
ENDIF
RETURN
END
```