Unclassfied

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER HIG-83-3 (Data Report 43)	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Archiving and Exchange of a Computerized Marine Seismic Database: The ROSE Data Archive System.		5. TYPE OF REPORT & PERIOD COVERED Final Report 6. PERFORMING ORG. REPORT NUMBER HIG-83-3 (Data Report 43)
7. AUTHOR(s) Sharon L. LaTraille		8. CONTRACT OR GRANT NUMBER(s) ONR N00014-82-C-0380
9. PERFORMING ORGANIZATION NAME AND ADDRESS Hawaii Institute of Geophysics 2525 Correa Road Honolulu, Hawaii 96822		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS --
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Ocean Sciences and Technology Division Bay St. Louis, MS 39520		12. REPORT DATE December 1983 13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Office of Naval Research Branch Office 1030 East Green Street Pasadena, CA 91106		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release: Distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Published as a Technical Report and Data Report by Hawaii Institute of Geophysics, University of Hawaii at Manoa, Honolulu, Hawaii 96822

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

ROSE Archive System
Data Archiving and Retrieval
ARCHIVING
Database Management
SEISMIC DATA STORAGE

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A central data exchange facility was organized at Hawaii Institute of Geophysics to manage seismic data collected during Project ROSE (the Rivera Ocean Seismic Experiment), a large marine seismic experiment. Prior to that experiment , wide consultation was made and much effort was expended in the establishment of a uniform, yet flexible, data exchange format. The participating institutions provided their data to the facility where the data were catalogued and distributed. This report describes in detail the

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

processes and computer programs used to catalog, store and distribute the ROSE seismic data. The users and I hope that exchange of data from other experiments will be facilitated by the use of the ROSE format and these programs.

DATA REPORT 43 HIG-83-3

ARCHIVING AND EXCHANGE OF A COMPUTERIZED MARINE SEISMIC DATABASE: THE ROSE DATA ARCHIVE SYSTEM

SHARON L. LaTRAILLE

DECEMBER 1983

Prepared for
OFFICE OF NAVAL RESEARCH
Contract N00014-82-C-0380
and
NATIONAL SCIENCE FOUNDATION
Grant EAR 80-02124

HAWAII INSTITUTE OF GEOPHYSICS
UNIVERSITY OF HAWAII
HONOLULU, HAWAII 96822

Data Report 43

HIG-83-3

ARCHIVING AND EXCHANGE OF A COMPUTERIZED MARINE SEISMIC DATABASE: THE ROSE DATA ARCHIVE SYSTEM

Sharon L. LaTraille

December 1983

Prepared for
OFFICE OF NAVAL RESEARCH
Contract N00014-82-C-0380

and

NATIONAL SCIENCE FOUNDATION
Grant EAR 80-02124

Charles E. Helsley
Director
Hawaii Institute of Geophysics

ABSTRACT

A central data exchange facility was organized at Hawaii Institute of Geophysics to manage seismic data collected during project ROSE (the Rivera Ocean Seismic Experiment), a large marine seismic experiment. Prior to that experiment, wide consultation was made and much effort was expended in the establishment of a uniform, yet flexible, data exchange format. The participating institutions provided their data to the facility where the data were catalogued and distributed. This report describes in detail the processes and computer programs used to catalog, store and distribute the ROSE seismic data. The users and I hope that exchange of data from other experiments will be facilitated by use of the ROSE format and these programs.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

I. Description of ROSE Experiment

A. Introduction

During early 1979 a consortium of universities and government agencies from the United States and Mexico conducted the Riveria Ocean Seismic Experiment (ROSE), a seismological and acoustical experiment off the western coast of Mexico. An overview of the three phases of the experiment, active (controlled source), passive (earthquake and microseismicity), and land based, is given by Ewing and Meyer [1982]. Twelve research groups deployed instruments at sea, and two groups deployed instruments on land.

Effective use of the large data set generated by the experiment required a data storage and exchange facility. The ROSE data exchange format was designed to enable exchange and processing on the various computer systems used by each research group. Because instrumentation for the experiment was not uniform, instrument parameters were defined and quantified in such a way that the data recorded by the instrument could be compared. Further, a concise identification scheme for both controlled-source and natural seismic events was developed. The data exchange and storage medium is industry-standard magnetic tape, and event-instrument pairs are catalogued on disk files. The combination of an easily read data exchange format and effective cataloging system has made the ROSE Data Exchange Facility viable. The Facility is based on the Harris H800 computer system at Hawaii Institute of Geophysics (HIG). A brief description of the exchange format and the operation of the Facility is given in LaTraille et al. [1982].

The data storage and retrieval procedures developed for the ROSE experiment are expected to be applicable to future experiments. This technical report contains complete listings of programs used, reports, forms, and detailed descriptions of procedures and the tape format. Samples of the tape header and file header records are shown in tabular form to provide the information associated with each instrument and each seismic event.

B. Size of the Data Base

The initial estimate of the number of events to be included in the data base was 180,000. An "event" is defined as one event recorded by one instrument. This number was based on detonating about 2000 shots to 76 OBS's during Phase I of the experiment, and 380 shots to 62 OBS's during Phase II. A 65% return rate of useful data was estimated. These calculations accounted for about 100,000 events. The other 80,000 were to be earthquakes recorded both on OBS's and land instruments. Programs and procedures were designed to handle this amount of data. As it turned out, the Exchange Facility was not asked to archive the land earthquakes, and the number of

useful events sent to the Facility was under 25,000 as of this report.

II. The Rose Storage and Exchange Format

A word about terminology: in this report we will use the terms "archive" and "storage" interchangably. The ROSE tape storage and exchange format was originally called the "Archive" format, but because it's main design feature is ease of data exchange between researchers and because it is not the most efficient format for data storage, it has been popularly referred to as the ROSE seismic data exchange format (see also LaTraille and Dorman, 1983). Nevertheless, both the Facility and the Format will be referred to with the term "Archive" in this report. A detailed description of the Format follows:

The storage and exchange medium is 9-track digital magnetic tape which has an external tape label containing the following information:

1) Instrument identification number.
2) Recording time window - this is the beginning and ending time covered by this tape.
3) Tape format, e.g., IBM NRZI @ 800 bpi or IBM PE @ 1600 bpi.
4) Tape length and thickness, e.g., 2400 ft., 1-1/2 mil.
5) ASCII or EBCDIC code.

The essential aspect of the data exchange format is to block the data into manageable segments and to identify the data attributes by a preceding header. The format is illustrated in Figure 1 and the contents of each record are described below.

A. The Tape Header File

The first file on the tape is a "tape header" file which provides descriptive information about the tape format, the data representation, and the instrument that generated the data. It is written in alphanumeric format because a large portion of the file contains text information. This file is actually a single 256-word record followed by an end-of-file mark. Table 1 illustrates the sample contents of a tape header. Each word group is described below:

Words	Contents
1 - 2	Instrument Identification Number. (A range of Instrument Identification numbers was assigned to each institution. The instrument description including response functions is on file with the data exchange center.)
3 - 30	Instrument designer's name and address, or originating institution name and address.
31 - 40	Experiment code, e.g., ROSE PHASE I.
41 - 45	Year, month, day, hour, and minute of start of data, e.g., 27 February 1979 @ 08:16 = 7902270816.

46 - 50 Year, month, day, hour and minute of end of data.

51 - 54 Number of files on tape exclusive of Tape Header (i.e., number of events).

55 -256 Not used.

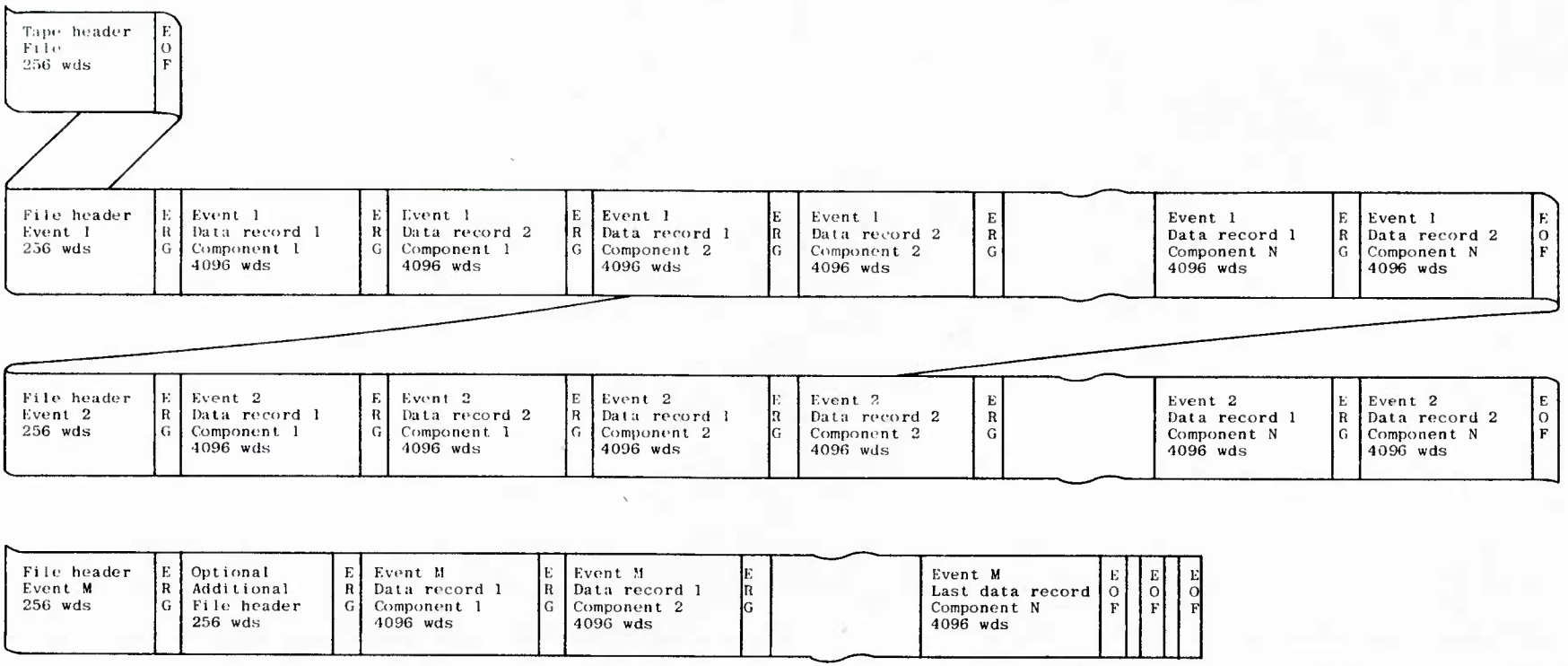


Figure 1. Block diagram of the ROSE digital tape data format for storage and exchange of seismic data. Events 1 and 2 illustrate the case where two records can fully define the event; event M can be fully described with one data record. An EOF is a hardware end-of-file mark. An ERG is an end-of-record gap. The tape header is in alphanumeric format (ASCII or EBCDIC). The data are in 16- or 32-bit twos-complement integer format (4096- or 2048-word records).

TABLE 1. Sample contents of tape header file*

Word Number	Field Description	Value
1-2	Instrument I.D. Number	302
3-30	Designer name and address	WA Prothero, UC Santa Barbara
31-40	Experiment code	ROSE Phase I
41-45	Year, month, day, hour and minute of start of data	7902020100
46-50	Year, month, day, hour and minute of end of data	7902051200
51-54	Number of files on tape exclusive of tape header (i.e., number of events)	107
55-256	Blank	

*All information alphanumeric (left justified).

B. File Header Record and Data Structure

The data section consists of a sequence of files each of which contains its own "file header record" followed by a data record(s) for each component, or channel, in the instrument. Optional additional headers can be used for instruments with more than 10 channels. The header is in integer format because almost all the data are numerical, and each header is 256 words long.

Each data record is 4096 words long. This size was chosen because most minicomputers can accommodate it in their I/O buffers. If a particular event requires more than 4096 words, the first 4096-word record is followed immediately with records 2, 3, ... until the event is described (see Figure 1). All samples from component 1 are written before going onto component 2. Thus, all of the data for an event are contained within a single file. Each file contains a number of data records equal to the number of components times the number of data records per component. If a data record contains less than 4096 samples, the record is padded with zeros. Words #71 and #72 in the header record are particularly important. Word #71 specifies the number of records required to fully describe an event for one component; word #72 specifies the number of non-zero samples in the last record of that component. The data are in twos-complement 16-bit integer format. Scales have been chosen so that a 16-bit word does not overflow. The format can also accommodate 32-bit integer data words, and in this case each record would contain 2048 data words. The sensitivities of the instruments, the electronics gains and the value of the least significant bit are specified in the header in order to preserve quantitative signal levels. The contents of the file header record are described below and a sample is shown in Table 2.

Words	Contents
1	Instrument Identification number. (This should be the integer equivalent of the identification number on the tape header so an instrument will have an alphanumeric and an integer representation on the same tape.)
2	Code for earthquake or shot data. If earthquake, set = 1; if shot, set = 2.
3	Earthquake or shot number. Shots are given specific numbers during an experiment. Numbers for earthquakes are simply a convenience--the true identifier for an earthquake is its time.
4	Year of first sample in file; 2-digit representation.
5	Month of first sample in file.
6	Day of first sample in file.
7	Hour of first sample in file.
8	Minute of first sample in file.

9 Seconds of first sample in file.

10 Millisecs of first sample in file. (All times UTC and corrected to WWV.)

11 Range in whole megameters between event and instrument.

12 Remainder of range in whole kilometers between event and instrument.

13 Remainder of range in whole meters between event and instrument.

14 Estimate of error in range in meters. As the method of computing the range may be subject to error and may vary according to method, information describing the calculation method and the sound speed data used is filed with the exchange center.

15 Instrument latitude in degrees.

16 Fractional part of instrument latitude in millidegrees.

17 Instrument longitude in degrees.

18 Fractional part of instrument longitude in millidegrees.

19 Instrument depth in meters.

20 Normal incidence (vertical) travel time from surface to instrument (millisecs).

21 Shot or earthquake latitude in degrees.

22 Fractional part of shot or earthquake latitude in millidegrees.

23 Shot or earthquake longitude in degrees.

24 Fractional part of shot or earthquake longitude in millidegrees.

25 Error radius in event location in meters. As the method of computing the event location may be subject to error and may vary according to the available information, e.g., satellite or LORAN for shots, some documentation describing the estimation method and the data used is filed with the data exchange center.

26 Shot depth in meters or epicenter depth in kilometers.

27 Water depth at shot or earthquake (in meters).

28 Year of shot or earthquake; 2-digit representation.

29 Month of shot or earthquake.

30 Day of shot or earthquake.

31 Hour of shot or earthquake.

32 Minute of shot or earthquake.

33 Second of shot or earthquake.

34 Millisec of shot or earthquake. (All times in UTC corrected to WWV.)

35 Code for explosive type used (1 = TOVEX, 2 = dynamite, 3 = tetratol, 4 = SUS (1.8 lbs.), 5 = maxipulse, 6 = airgun, 7 = T.N.T., 8 = open, 9 = other).

36 Shot weight in 1000 $\log_{10}$ W, where W is charge weight in grams; or 10M where M is the earthquake magnitude.

37 Bubble pulse period in millisecs.

38 Sampling rate in samples/sec.

39 Number of digitized components.

40 Number of data words in record (= 4096 for 16-bit words or 2048 for 32-bit words).

41 File number within tape.

42-59 Blank for additions and changes.

60 Code for duplicate component parameters - 1 if all channels have same parameters as channel 1. If channel parameters individually coded, this code = 0.

61 Code for component number one, e.g., 1) vertical, 2) radial, 3) tangential, 4) hydrophone, 5) non-rotated first horizontal, 6) non-rotated second horizontal, 7) water wave channel, 8) time.

62 Azimuth of non-rotated first horizontal if 61 is (5) or (6).

63 Sensitivity of instrument at frequency f_0. (For displacement instruments - milli volts/millimicron; for velocity instruments - milli volts/(cm/sec), or volts (m/sec), and for hydrophones - volts/micropascal. The detailed response curve of each instrument is on file with the exchange center.)

64 f_0 in milliHertz - the frequency at which the above sensitivity of the instrument was measured.

65 Lower cutoff of passband in instrument amplifier (Hz).

66 Upper cutoff of passband in instrument amplifier (Hz).

67 Amplifier gain of digitizing filter in dB.

68 Lower cutoff of passband of digitizing filter (Hz).

69 Upper cutoff of passband of digitizing filter (Hz).

70 Value of the low order, or least significant, bit of the 16- or 32-bit integer word in microvolts.

71 Number of records in component.

72 Number of samples in last record.

73-80 Blank for changes and additions.

81-100 Repeat content of words 61-80 for the second component.

101 Continue repeating at intervals of 20 words for each component in the instrument as required up to 10 components.

If channel parameters for more than 10 components (channels) are to be encoded, add as many additional 256-word records as needed (one for each 12 additional components). Encode the contents of words 61-80 for the 11th component in words 1-20 of the new header record. Continue repeating at intervals of 20 words for each additional component as required. Note: if additional event header records are used, there must be a 0 in word 60 of the main event header.

TABLE 2. Sample Contents of File Header Record

Word Number	Field Description	Value
1	Instrument I.D. Number.	5
2	Code for earthquake or shot data. If earthquake, set=1, if shot, set=2.	2
3	Earthquake or shot number.	3243
4	Year of first sample in file.	79
5	Month of first sample in file.	2
6	Day of first sample in file.	28
7	Hour of first sample in file.	0
8	Minute of first sample in file.	5
9	Second of first sample in file.	9
10	Millisecond of first sample in file.	310
11	Range in whole megameters (between event and instrument).	0
12	Remainder of range in whole kilometers.	24
13	Remainder of range in whole meters.	368
14	Estimate of error in range in meters.	500
15	Instrument latitude in degrees.	15
16	Fractional part of instrument latitude in millidegrees.	29
17	Instrument longitude in degrees.	-104
18	Fractional part of instrument longitude in millidegrees.	-750
19	Instrument depth in meters.	2476
20	Vertical travel time from surface to instrument (msec).	1692
21	Event latitude in degrees.	15
22	Fractional part of event latitude in millidegrees.	253
23	Event longitude in degrees.	-104
24	Fractional part of event longitude in millidegrees.	-756
25	Error radius in event location in meters.	0
26	Shot depth in meters; or epicenter depth in kilometers.	41
27	Water depth at event in meters.	2681
28	Year of shot or earthquake.	79
29	Month of shot or earthquake.	2
30	Day of shot or earthquake.	28
31	Hour of shot or earthquake.	0
32	Minute of shot or earthquake.	5
33	Second of shot or earthquake.	9
34	Millisecond of shot or earthquake.	342
35	Code for explosive type used (1 = TOVEX, 2 = dynamite, 3 = tetratol, 4 = SUS (1.8 lbs.), 5 = maxipulse, 6 = airgun, 7 = TNT, 8 = open, 9 = other).	3
36	Shot weight in 1000 $\log_{10}$W, where W is charge weight in grams; or 10M where M is the earthquake magnitude.	3434
37	Bubble pulse period in milliseconds.	110
38	Sampling rate in samples/second.	100
39	Number of digitized components.	2
40	Number of data words per record.	4096
41	File number within tape.	2
42-59	Blank.	

C. Shot Instant Data

In order for a participant to complete the header information for data recorded on his instruments from shots detonated by another participant, the former must have shot instant times, shot locations and other information pertinent to seismic data interpretation. These data were distributed to participants on magnetic tape. The file consisted of one 81 character record per shot. The variables and their Fortran format are listed below:

Variable Name	Format (FORTRAN)	#Characters
Shot Number	I4	4
Shot Instant: Year; 2-digit representation	I3	3
Month	I3	3
Day	I3	3
Hour	I3	3
Minute	I3	3
Second	F7.3	7
Latitude	F10.4	10
Longitude	F10.4	10
Size (KG)	F10.3	10
Shot Depth (M)	F6.1	6
Water Depth (MSEC)	I6	6
Bubble Pulse Period (MSEC)	I5	5
Explosive Type	I2	2
Shot Instant Correction	F6.3	6

Although this format has not been very widely discussed and accepted as the ROSE data storage and exchange format, it is easily readable and probably should be an addendum to the exchange format.

III. Archival and Retrieval Procedures

A. Archiving the Data

ROSE participants send their data to the Exchange facility on 9-track 800 or 1600-bpi digital magnetic tape in the data exchange format. The tape header file and the event header records from each tape become part of the "Catalog" area on the disc. This area is the main source for reports on the data base and for request documentation. The Catalog has the capability of being easily sorted by ´keywords´, and it is backed up on magnetic tape.

A simplified picture of archiving is shown on the flow chart in Figure 2. A data tape is received, logged in and its receipt is acknowledged. Any accompanying documentation is filed. Next the tape is read onto disc and verified as to format. Archiving information is added to the tape header file (descriptor file). Then the headers and data are written to an archive tape in ROSE format. The header information is, at the same time, merged with the Catalog and an archiving summary report is printed (ROSE Archive Report #1, Appendix A). The archive tape is then verified, labelled, logged and stored. The original tape is recycled for use in the data archive system. The format of the data stored on ROSE Archive tapes will be identical to the official ROSE data archive format; however, tapes will have an additional Exchange

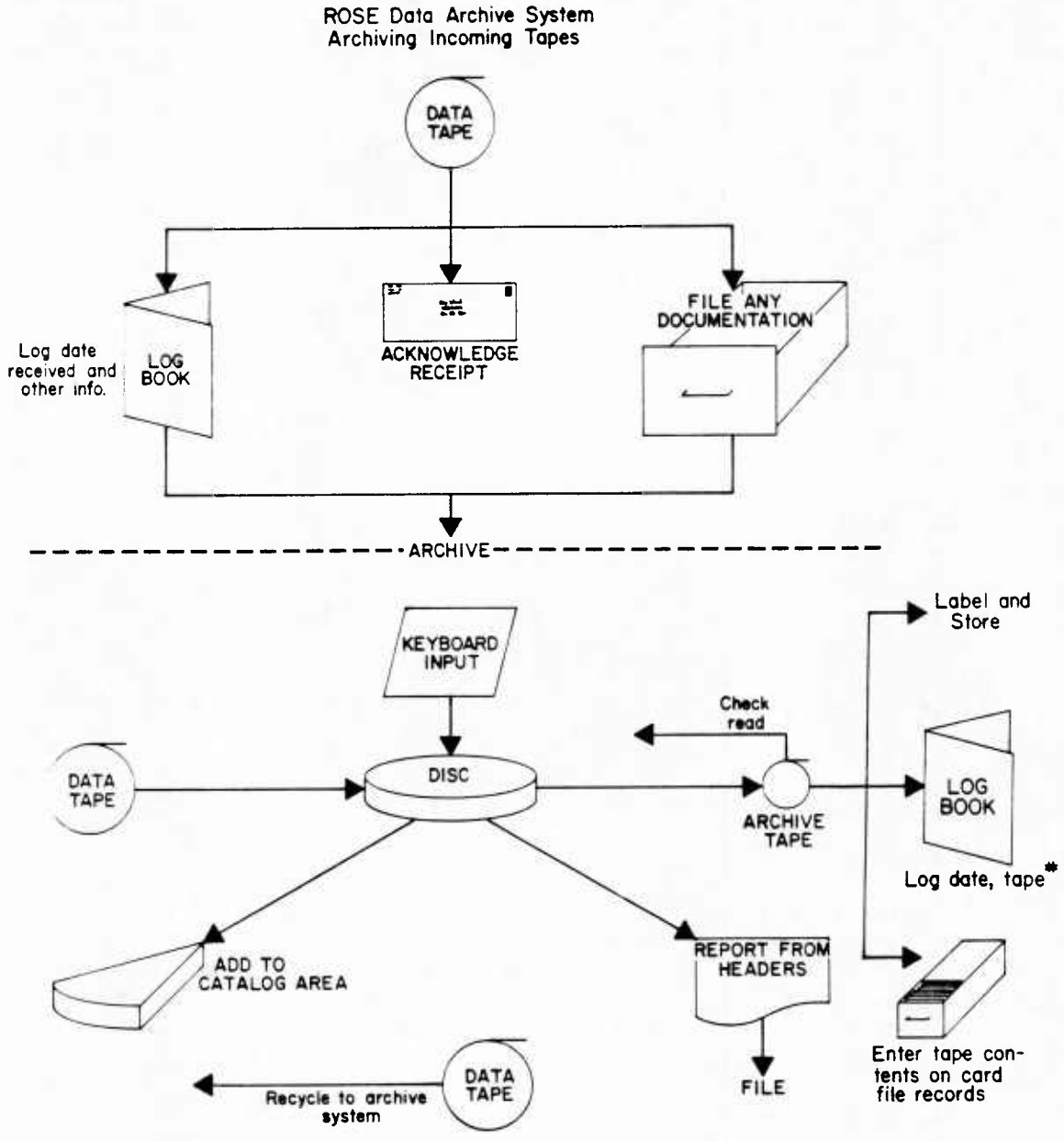


Figure 2. Flow chart of the procedures for processing and storage of digital seismic data.

Facility tape header at the beginning which will contain the following information:

1. Tape # in form of RARCnnnn, where n's are integers
2. Date archived
3. Date received source tape
4. Institution received from
5. Key for documentation on file
6. Keyword value ranges

This gives us a double check system on the archive tapes, and makes determination of what is on the tape simple and efficient. A sample of the contents of an Exchange Facility tape header is shown in Table 3.

1. Keywords

A "KEYWORD" is defined as an attribute of the data, contained in either the tape header file or the file header record, which can be used to define the portion of the ROSE DATA BASE requested by a user. The "keyword" may be described by more than one word in the header i.e., time is actually 5 words in the tape header, but is a single data attribute and thus a single keyword. We have defined 18 keywords as shown in Table 4.

TABLE 3. Sample Contents of Exchange Facility Tape Header

DESCRIPTION	SAMPLE CONTENTS
ROSE Archive Tape Number	77
Shot Line Number	SLN6S
Institution Rec'd tape from	HIG; Gettrust
Documentation Code; 1=YES	1
Data Archive Tape Received	30 JUL 81
Data Archived	07 AUG 81
Date Last Updated	20 APR 82
Date Last Accessed	15 DEC 82
Instrument Number	503
Event Numbers	3134 3241
Minimum Data Start Time	1979 2 10 16 59 56 792
Maximum Data Start Time	1979 2 11 02 00 20 928
Explosive Types	0 0
Water Depth at Instrument	2955
Water Depths at Event	2523 3122
Instrument Depth (2 way msec)	1970
Event Depth	36 73
Number of Channels	4 4
Event Types	2 2
Event Sizes	2.3 11.3
Ranges	2.9 79.5
Instrument Latitude	11.4114
Instrument Longitude	-103.5133
Event Latitutde Range	11.293 11.392
Event Longitude Range	-103.504 -104.233

TABLE 4. Keywords and their Definitions

KEYWORD	DESCRIPTION	KEYWORD	DESCRIPTION
1. ENUM	Event number	10. STIME	Data start time
2. INUM	Instrument I. D. no.	11. RANGE	Range-event to instrument
3. ELAT	Event latitude	12. ILAT	Instrument latitude
4. ELON	Event longitude	13. ILON	Instrument longitude
5. TYPE	Event type	14. IDEP	Instrument depth
6. EDEP	Event depth	15. WDEPI	Water depth at instrument
7. WDEPE	Water depth at event	16. CHAN	Channel or component number
8. SIZE	Event size	17. SHOTLN	Shotline designation
9. EXPL	Explosive type	18. TAPE	Storage tape number

2. The Disc Catalog format

The Catalog file is an unblocked, direct access area stored on the Harris H-800 disc. It contains most of the information in the event headers of all events stored in the Archive, and it contains all the tape headers describing the tapes in storage. The event header information is stored in binary format and the tape headers are ASCII. The structure of the file is as follows:

(3) 112 word records, the last of which contains the address of the first tape header
(1) EOF
(N) Event header records
(1) EOF
(M) 224 word tape headers, each of which contain the address of the first event header record on that tape

Each time tape headers and event headers are added to the Catalog, the starting position of the header records changes. This starting position is updated and written onto record 3 at the time each tape is archived. The format for the tape headers is shown in Table 3. For more efficient storage and ease of manipulation of data attributes such as event time, location, range and size, the event header information is stored in the Catalog file as shown in Table 5.

Table 5. Binary Event Records

Word #	Variable	Type	Description
1	ITYPE	INT	Record type
2	TAPENM	INT	Tape number
3	INUM	INT	Instrument number
4	ENUM	INT	Event number
5-6	STIME	INT*6	Data start time; c.msec
7-8	SBT	INT*6	Event time
9-10	SIZE	REAL	Event size; kg
11-12	RANGE	REAL	Event to instrument range, km
13-14	ILAT	REAL	Instrument latitude
15-16	ILON	REAL	Instrument longitude
17-18	ELAT	REAL	Event latitude
19-20	ELON	REAL	Event longitude
21	EXPLOS	INT	Explosive type
22	WDEPI	INT	Water depth at instrument; msec
23	WDEPE	INT	Water depth at event; m
24	IDEP	INT	Instrument depth; m
25	EDEP	INT	Event depth; m
26	ICHN	INT	# of channels
27	TYPE	INT	Event type code
28	RERR	INT	Error estimate in range; m
29	ELOER	INT	Error radius in event location; m
30	BUB	INT	Bubble pulse period; msec
31	SAMP	INT	Sampling rate; samp/sec
32	NWDS	INT	Number of words/rec (4096)
33	FNUM	INT	File number within tape
34	NREC	INT	Number of recs/component
35	NSAMP	INT	Number of samples in last record
36	IDEL	INT	Delete code; 1=delete

3. Modifications to Data Archived

Modifications to the event and instrument attributes are made in the computer Catalog file only; the original archived data is not changed. These modifications are incorporated in retrieved data and reports, and the fact that it has been modified is noted. Major errors are corrected by the originating institution, which sends a new tape to the Exchange Facility.

B. Retrieval and Exchange

Special order forms have been designed to facilitate processing requests sent to the ROSE Data Archive System. One is the Shot Request Form, another is the Earthquake Request Form. Samples of these forms appear in Appendix A.

An overview of the data (or report) request handling and retrieval procedures follows (see flow chart in Figure 3):

A request is received, logged in and acknowledged. Each request is entered via computer terminal by means of an interactive PASCAL program using the archive language of keywords. The request is then translated into a unique FORTRAN program which handles searching, sorting, merging and reporting. Entering and processing a request is made extremely simple to the user. The search and retrieval programs were designed to process intricate requests for portions of the data base. In practice, this capability has not been used to the fullest possible extent. Most investigators have requested whole shotlines, for example, and then selected data useful to them at their own facilities. In addition to producing FORTRAN source, the initial program writes information on the request into a storage area for status reporting and prints a summary of the request. The request can be processed at any time by compiling the FORTRAN source. The result of this is 1) a report of the location (i.e., tape numbers or disc file names) of requested data and notes on any special documentation available (ROSE Archive Report #3, Appendix A); and 2) a file of event headers containing any corrections to be incorporated into the data to be exchanged. At this point, if the request was merely for a printed report on the amount of data available which fit the specified limits, the process would be finished, except for mailing the report. The usual request will be for the data itself, and in this case the next step is the actual retrieval. Using the list of tape and file numbers, a data technician will read in the data, merge updated event headers with the data, sort it into the requested order and write it out in ROSE format with a tape header file describing the tape contents. After being verified, the tape or tapes will be logged and then sent to the requesting participant along with available documentation and a summary report of the data.

Bi-annual bulletins are sent to participants in the ROSE experiment to keep them appraised of data availability at the archive. A sample bulletin is contained in Appendix B.

IV. Computer Hardware and Software for the Archive

A. Hardware

The ROSE Data Archive uses the H.I.G. computer facility which also serves as the primary scientific computer system for faculty, researchers

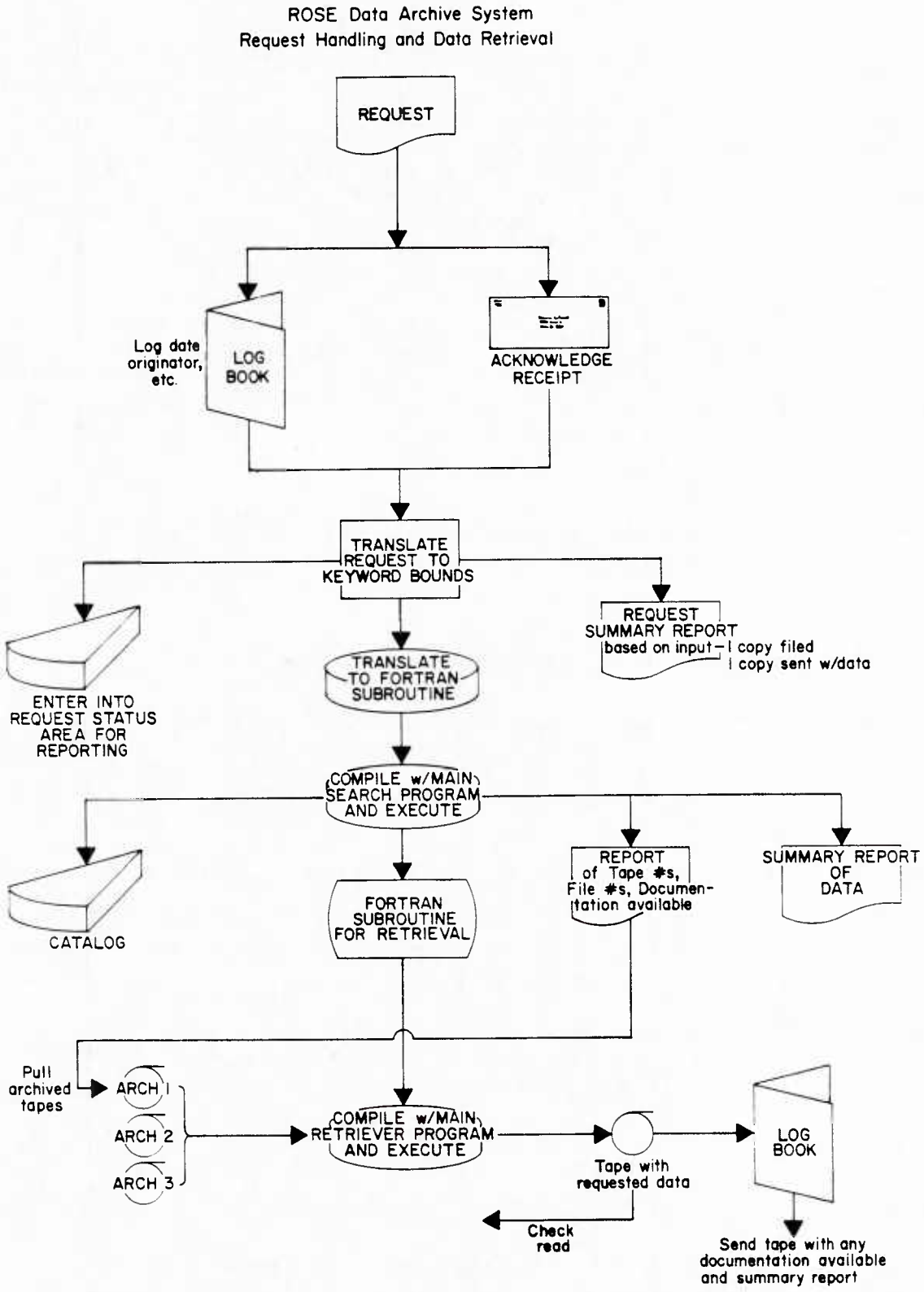


Figure 3. Flow chart of the procedures for the retrieval of digital seismic data.

and students associated with the Institute. The facility's equipment is based on a Harris H800 computer with virtual memory access, with 448k words of MOS memory and three 300 Megabyte disc modules. See Appendix C for a complete list of peripherals and software. The ROSE Archive is but one of a number of "users" of the computer system where 64 terminals and 15 background programs compete for system resources. To minimize the impact of Archive data handling in such a competitive system, the computer programs for the Archive were written to be input-output (I/O) limited. This has turned out to be reasonably simple since the combination of PASCAL and Fortran 77 languages allows us to generate very efficient code for all the tasks which are a part of the Archive. Due to the multiple, independent I/O channel design of the Harris computer system and efficient programming, the impact on research computing has been minimal. In fact, the additional disc storage unit and two magnetic tape drives purchased for the ROSE Archive benefit research projects when the Archive does not require their use.

B. Software

The following is a list of program names and descriptions of their function. Program listings and examples of input and output appear in Appendix D.

Programs Used By ROSE Archive & Retrieval System

Source Name	Description/Function
DISTAP	-Initial check of incoming archive tapes reads and displays to screen archive tape header and event headers -Can also display an HIG archived tape, i.e., with an HIG header.
DVIEW	-Displays first 50 data values of N records on an incoming tape.
RECLEN	-Reads incoming (or any) tape and prints record lengths and event numbers.
BIARCH	-Reads incoming tape, adds info to Catalog, builds HIG archive header, archives to new tape. Auxiliary programs: M<BIARC, HEDGEN, TAPOS.
BHIROS	-Same as BIARCH for data in the format used internally at HIG. Auxiliary: HRAMAC, ROSEHD, HEDGEN, TAPOS.
DISCAT	-Display Catalog tape headers and/or file header records.
LISHDR	-List the Catalog tape headers either by tape or instrument.
RDHDR	-List Catalog event headers.
FIX	-Edit Catalog event headers.

ROSED	-Edit Catalog tape headers.
ADHED, ADTAPE, DELHDR, DELTAP	- Add and/or delete Catalog records.
BULLETIN	-Generate "Bulletin" from Catalog after sorting.
SEARCH	-Search Catalog for storage tape location of specified instruments, events, times, positions, etc. Auxiliary programs RECSEL, SELECT, M<SEARCH.
RETREV	-Retrieve specified data from archive tapes. Auxiliary program M<RETREV.
RARHIG	-Convert data in ROSE Archive format to internal HIG format.
ITMCNT	-Time conversion subroutine used in all programs. Entry points CNTITM, LJLCNT.

V. Summary

The ROSE participants have gathered a large, unique marine and land seismic database which, by prior agreements, is being shared through a data storage and exchange facility located at the Hawaii Institute of Geophysics. The data are available to all participants and to the entire scientific community. After 1983, these data will be available from National Geophysical Data Center, Boulder, Colorado. The combination of an easily read data exchange medium and effective cataloging system has made the ROSE Data Exchange Facility viable.

The ROSE project represents one of the new types of data intensive marine programs that will be pursued and extended in the future. It is anticipated that the formats developed for this experiment will be the standard for storing and exchanging marine refraction data.

ACKNOWLEDGMENTS

I thank all of the participants in the ROSE experiment, especially John Ewing, who coordinated the entire project. I appreciate the help of A. Baggeroer and P. Stoffa, who designed and tested the data storage and exchange format. I thank Joseph F. Gettrust for sound guidance in the early phases of operation of the exchange facility. Marie Harris typed the report, Rita Pujalet and Barbara Jones edited and prepared the report for publication. The ROSE data storage and exchange facility is funded by the Office of Naval Research and by the National Science Foundation under grant EAR80-02124.

REFERENCES

Ewing, J.I. and Meyer, R.P., 1982, Rivera Ocean Seismic Experiment - ROSE Overview, J. Geophys. Res., 87, p. 8345-8358.

LaTraille, S.L., Gettrust, J.F., and Simpson, M.E., 1982, The ROSE Seismic Data Storage and Exchange Facility, J. Geophys. Res., 87, p. 8359-8363.

LaTraille, S.L. and Dorman, L.M., 1983, A Standard Format For Storage and Exchange of Natural and Explosive-Source Seismic Data: the ROSE Format, Mar. Geophys. Res., 6, p. 99-105. Note: this reference is included for information only.

APPENDIX A

Forms and Reports

FORM #1

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

SEND TO: SHARON LATRAILLE, ROSE ARCHIVE
HAWAII INSTITUTE OF GEOPHYSICS
2525 CORREA ROAD
HONOLULU, HAWAII 96822

ROSE DATA ARCHIVE SHOT REQUEST FORM

PLEASE STATE AS CONCISELY AS POSSIBLE WHAT REPORT IS TO BE GENERATED OR WHAT DATA IS TO BE RETRIEVED____

____

____

____

PLEASE DESCRIBE YOUR REQUEST IN DETAIL BY CHECKING OR FILLING IN THE FOLLOWING SHOT AND INSTRUMENT DESCRIPTORS. CHECK THE "ALL" BOX TO INDICATE YOU WANT ALL SHOT/INSTRUMENTS PAIRS WITH THAT CHARACTERISTIC.

___REPORT ONLY

___REPORT AND DATA RETRIEVAL

9 TRACK TAPE PARAMETERS:
DENSITY: 800 ( ) OR 1600 ( ) BPI
FORMAT: ASCII( ) OR EBCDIC ( )

FOR EACH DATA REQUEST YOU WILL RECIEVE A LISTING DESCRIBING THE DATA BEING SENT, ANY DOCUMENTATION AVAILABLE ON THE INSTRUMENTS INVOLVED, AND THE DATA ITSELF IN ROSE FORMAT ON 9 TRACK TAPE.

****

SHOT DESCRIPTORS

SHOT NUMBERS: ALL ( ), OR SPECIFY #S____

SHOT TYPE: ALL ( ), OR SPECIFY TYPE #S____________________________________
1)TOVEX, 2)DYNAMITE, 3)TETRATOL, 4)SUS, 5)MAXIPULSE, 6)AIRGUN, 7)TNT, 8)OPEN, 9)OTHER

SHOT LINES: ALL ( ), OR SPECIFY LINE #S ___________________________________

TIME WINDOW: ALL ( ), OR SPECIFY FROM__________________Z TO______________Z

SHOT SIZE: ALL ( ), OR SPECIFY SIZE BOUNDS(LBS)_____________________________

LOCATION: ALL( ), OR SPECIFY FROM LATITUDE_____________TO LATITUDE___________
AND FROM LONGITUDE_______________TO LONGITUDE___________________

WATER DEPTH: ALL( ), OR SPECIFY DEPTH BOUNDS(M)_____________________________

SHOT DEPTH: ALL( ), OR SPECIFY DEPTH BOUNDS(M)______________________________

RANGE, EVENT TO INSTRUMENT: ALL( ), OR SPECIFY BOUNDS(KM)__________________

INSTRUMENT DESCRIPTORS

I.D. NUMBER: ALL( ), OR SPECIFY NUMBERS_____

LAND VERSUS SEA STATION: ALL( ), OR CHECK ONE: OCEANIC( ) CONTINENTAL( )

LOCATION: ALL( ), OR SPECIFY FROM LATITUDE_____TO LATITUDE_____
AND FROM LONGITUDE_____TO LONGITUDE_____
INSTRUMENT DEPTH: ALL( ), OR SPECIFY BOUNDS(M)_____
WATER DEPTH: ALL( ), OR SPECIFY BOUNDS(MSEC)_____
COMPONENT (CHANNEL) #S: ALL( ), OR SPECIFY CODE #S_____
1)VERTICAL, 2)RADIAL, 3)TANGENTIAL, 4)HYDROPHONE, 5)NON-ROTATED FIRST HORIZONTAL, 6)NON-ROTATED SECOND HORIZONTAL, 7)WATER WAVE, 8)TIME

FORM #2

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

SEND TO:
SHARON LATRAILLE, ROSE ARCHIVE
HAWAII INSTITUTE OF GEOPHYSICS
2525 CORREA ROAD
HONOLULU, HAWAII 96822

ROSE DATA ARCHIVE EARTHQUAKE REQUEST FORM

Please state as concisely as possible what report is to be generated or what data is to be retrieved____
____
____
____

Please describe your request in detail by checking or filling in the following earthquake and instrument descriptors. Check the All box to indicate you want all earthquake/instrument pairs with that characteristic.

___ Report only

___ Report and data retrieval

9 track tape parameters:
Density: 800 ( ) or 1600 ( ) BPI
Format : ASCII( ) or EBCDIC( )

For each data request you will receive a listing describing the data being sent, any documentation available on the instruments involved, and the data itself in ROSE format on 9 track tape.

****

EARTHQUAKE DESCRIPTORS

Earthquake #s: ALL( ), OR SPECIFY CATALOG #S______________________________

Recording time window: ALL( ), OR SPECIFIY FROM______________________Z TO ______________________Z

Magnitude: ALL( ) OR SPECIFY BOUNDS____

location: ALL( ) OR SPECIFY FROM LATITUDE____________TO LATITUDE__________ __________AND FROM LONGITUDE________________TO LONGITUDE_________________

Land versus sea: ALL( ), OR CHECK ONE: OCEANIC ( ) CONTINENTAL ( )

Limit in location error: ALL( ), OR SPECIFY ERROR RADIUS BOUNDS(M)_______

Source depth: ALL( ), OR SPECIFY DEPTH BOUNDS(KM)___________________________

Range, event to instrument: ALL( ), OR SPECIFY BOUNDS(KM)_________________

INSTRUMENT DESCRIPTORS

I.D. Number: ALL( ), OR SPECIFY NUMBERS____________________________________

Land versus sea station: ALL( ), OR CHECK ONE: OCEANIC( ) CONTINENTAL ( )

Location: All( ), OR SPECIFY FROM LATITUDE_____TO LATITUDE_____
AND FROM LONGITUDE_____TO LONGITUDE_____

Instrument Depth: ALL( ), OR SPECIFY BOUNDS(M)_____
Water depth: ALL( ), OR SPECIFY BOUNDS(MSEC)_____
Component (Channel) #S: ALL( ), OR SPECIFY CODE #S_____
1)Vertical, 2)Radial, 3)Tangential, 4)Hydrophone, 5)Non-rotated first horizontal, 6)Non-rotated second horizontal, 7)Water wave, 8)Time

SEND REPORT AND/OR DATA TO:_____

FORM #3

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>

SELECT PROGRAM INPUT FORM

1. INSTRUMENT # (INUM)________	10. EXPLOSIVE TYPE (EXPLOS)________ (1=TOVEX,2=DYN,3=TET) (4=SUS,5=MAXI,6=AG) (7=TNT,9=OTHER)
2. EVENT # (ENUM)________	11. WATER DEPTH @ INST. (WDEPI)________ (ONE-WAY MSEC)
3. DATA START TIME (STIME)________ (YR,MO,DA, HR,MIN,SEC) USE COMMAS	12. WATER DEPTH AT EVENT (WDEPE)________ (METERS)
4. EVENT SIZE (SIZE)________ (SHOTS:KG;QUAKES: 10*MAGNITUDE OR CODA LENGTH,SEC*10)	13. COMPONENT TYPE (CHAN)________ (1=VERT,2=RAD,3=TANG) 4=HYDRO,5=1ST HORIZ) (6=2ND HOR, 7=WW,8=TIME)
5. RANGE (KM) (RANGE)________	14. EVENT DEPTH (EDEP)________ (METERS)
6. INST. LATITUDE (ILAT)________	15. INST. DEPTH (IDEP)________ (METERS)
7. INST. LONGITUDE (ILON)________	16. EVENT TYPE (TYPE)________ (1=QUAKE,2=SHOT)
8. EVENT LATITUDE (ELAT)________	17. SHOTLINE (SHOTLN)________ (LIKE 1S,3L,1T3)
9. EVENT LONGITUDE (ELON)________	18. ARCHIVE TAPE NO. (TAPENM)________ (4 DIGITS)

VALID LOGICAL SYMBOLS: /-TO, =-EQUALS,
<-LESS THAN, >-GREATER THAN,
-NOT EQUAL, BLANK-AND

Example: ENUM 505/510 means event #s 505 to 510
RANGE >500 means range greater than 500
TYPE 1 2 means type 1 and 2

Type SELECT and then enter the select input information from this form using logical symbols, if needed. Enter 1 or more lines. To stop entering, type END and 2 carraige returns. List RECSEL and check to see if it is correct. Then type SEARCH and the following will happen:

a) Subroutine will be compiled
b) Program will be vulcanized
c) Catalog will be searched

ROSE ARCHIVE REPORT #1

* * * SUMMARY OF DATA ARCHIVED * * * RUN DATE, TIME
21 MAY 80, 11:11:43

RARC TAPE # 17 DATE RECEIVED: 19 MAY 80 EXPERIMENT: ROSE I
TAPE DATA START TIME: 79 2 3 21 00
TAPE DATA END TIME : 79 2 4 8 00

INSTRUMENT #: 505 DESIGNER: HIG ENGR., HAWAII INST GEOPHYSICS
DOCUMENTATION CODE (YES=1): 1 INSTITUTION RECEIVED FROM: H.I.G.; GETTRUST
COMPONENTS 1-10 ONLY: H1P TIV INSTRUMENT LAT,LON: 11.8339,-102.7839
INSTRUMENT DEPTH: 3096 M. WATER DEPTH AT INSTRUMENT: 2064 MSEC.
NUMBER OF EVENTS: 35

KEYWORD MINIMUM & MAXIMUM VALUES:

EVENT TYPE	2				SHOT LINE #:	SLN3L						
EVENT #S					1007	TO	1041					
DATA START TIMES	1979	2	3	20	59	TO	1979	2	4	7	59	
EVENT LATITUDES				11.027		TO	12.943					
EVENT LONGITUDES				-102.587		TO	-102.913					
EVENT DEPTHS				57		TO	132 M					
WATER DEPTHS				2790		TO	3168 M					
EVENT SIZES				87.1		TO	232.2 KG					
EXPLOSIVE TYPE				1								
RANGES-EVENT TO RCVR				10.1		TO	124.1 KM					

ROSE ARCHIVE REPORT #2

ROSE ARCHIVE TAPE NO.	1
SHOT LINE NO.	SLN1T
INSTITUTION RECD TAPE FROM	MASS. INSTITUTE OF TECHNOLOGY
DOCUMENTATION CODE; 1=YES	1
DATE ARCHIVE TAPE RECEIVED	26 FEB 80
DATE ARCHIVED	5 MAR 80
DATE LAST UPDATED	5 MAR 80
DATE LAST ACCESSED	5 MAR 80
**********MINIMUM & MAXIMUM VALUES OF KEYWORDS**********	
INSTRUMENT NUMBER	1001
EVENT NUMBERS	1001 5002
MINIMUM DATA START TIME	1979 2 2 2 0 0 0
MAXIMUM DATA START TIME	1979 2 2 11 59 50 0
EXPLOSIVE TYPES	1 1
WATER DEPTH AT INSTRUMENT	676
WATER DEPTHS AT EVENT	2970 6715
INSTRUMENT DEPTH	1014
EVENT DEPTHS	100 133
#S OF CHANNELS	12 12
EVENTS TYPES	2 2
EVENT SIZES	816.5826 907.8208
RANGES	39.8110 347.3430
INSTRUMENT LATITUDE	12.2400
INSTRUMENT LONGITUDE	-101.9570
EVENT LATITUDE RANGE	10.3470 14.0050
EVENT LONGITUDE RANGE	-101.2880 -104.6060

***** TAPE HEADER FILE CONTENTS *****

INSTR. # 1001 DESIGNER NUSC NEW LONDON LAB NEW LONDON, CONN 06320

EXPERIMENT: ROSE;MABS # OF EVENTS: 11

TAPE DATA START TIME	1979 2 2 2 0 0 0
TAPE DATA END TIME	1979 2 2 12 0 0 0

ROSE ARCHIVE REPORT #3

* * * DATA RETRIEVAL REQUEST * * *

REQUEST RECEIVED: 9/5/79
REQUESTING PARTY: J. GETTRUST, HIG

REQUEST KEYWORD BOUNDS (KEYWORD BOUNDS="ALL" NOT LISTED)

INSTRUMENT NUMBERS:	514 TO 516
EVENT NUMBERS:	4050 TO 4555
EVENT TYPE:	2
EVENT SIZES	30 TO 200
COMPONENTS:	V P

DATA LIST

RARC TAPE NUMBER	INSTRUMENT	EVENTS
29	514	4050 - 4128
30	514	4129 - 4320
31	514	4321 - 4440
32	514	4441 - 4555
40	515	4200 - 4320
41	515	4321 - 4420
42	515	4421 - 4555
55	516	4495 - 4555

ROSE ARCHIVE REPORT #4

* * * SUMMARY OF DATA RETRIEVED * * *

REQUEST RECEIVED: 9/9/79
REQUESTING PARTY: JOE GETTRUST

REQUEST KEYWORD BOUNDS (KEYWORDS BOUNDS="ALL" NOT LISTED)

INSTRUMENT NUMBERS:	511 TO 514
EVENT NUMBERS:	4098 TO 4138
EVENT SIZE:	2
COMPONENTS:	200 TO 5000

DATA RETRIEVED (IN THE ORDER ON THE TAPE)

TAPE NUMBER 1

INSTRUMENT #	EVENT #	TYPE	SIZE(KG)	LATITUDE	LONGITUDE	COMPONENTS
511	4098	2	816.47	12.4918	-100.9702	V P
511	4099	2	816.47	12.5039	-101.2876	V P
511	4100	2	816.47	12.3212	-101.6007	V P
511	4128	2	81.65	11.9644	-101.8834	V P
511	4129	2	81.65	11.9249	-101.8747	V P
511	4130	2	81.65	11.8856	-101.8661	V P
511	4131	2	81.65	11.8463	-101.8578	V P
511	4132	2	81.65	11.8072	-101.8499	V P
511	4133	2	217.73	11.7695	-101.8420	V P
511	4134	2	217.73	11.6914	-101.8251	V P
511	4135	2	217.73	11.6138	-101.8086	V P
511	4136	2	217.73	11.5363	-101.7921	V P
511	4137	2	217.73	11.4585	-101.7752	V P
511	4138	2	217.73	11.3811	-101.7589	V P
512	4098	2	816.47	12.4918	-100.9702	V P
512	4099	2	816.47	12.5039	-101.2876	V P
512	4100	2	816.47	12.3212	-101.6007	V P
512	4101	2	816.47	13.4234	-102.1361	V P
512	4102	2	816.47	13.7728	-102.2160	V P
512	4103	2	816.47	14.0833	-102.2561	V P
512	4128	2	81.65	11.9644	-101.8834	V P
512	4129	2	81.65	11.9249	-101.8747	V P
512	4130	2	81.65	11.8856	-101.8661	V P
512	4131	2	81.65	11.8463	-101.8578	V P
512	4132	2	81.65	11.8072	-101.8499	V P
512	4133	2	217.73	11.7695	-101.8420	V P
512	4134	2	217.73	11.6914	-101.8251	V P
512	4135	2	217.73	11.6138	-101.8086	V P
512	4136	2	217.73	11.5363	-101.7921	V P
512	4137	2	217.73	11.4585	-101.7752	V P
512	4138	2	217.73	11.3811	-101.7589	V P

APPENDIX B

SAMPLE ARCHIVE STATUS REPORT


Archive Status Report
As of
6/30/82
HAWAII INSTITUTE OF GEOPHYSICS

Rose Archive Bulletin
Volume 3 Number 1
06/30/82

Since the December 1981 edition of the Archive Status Bulletin, the volume of data sent to the storage facility has leveled off. Most of the Phase I shot data is now on archive tapes. Oregon State University added over 2000 Phase I shots early this year. The ROSE data storage and exchange facility now houses about 21,600 events (event-instrument pairs) , with eight participating institutions represented. They are Hawaii Institute of Geophysics, Oregon State University, Scripps Institute of Oceanography, Woods Hole Oceanographic Institute, Massachusetts Institute of Technology, University of Texas Marine Science Institute, University of California at Santa Barbara and University of Washington. The data from Lamont-Doherty Geological Observatory is expected this summer. A listing of the data contained in the Archive, including earthquake data, is attached along with a summary of the events for each participating institution.

Several large requests for data have been processed since the last report. Request handling has been the main activity; our retrieval procedures have smoothed out considerably so that time to process a request now averages one week.

An article entitled "The ROSE Seismic Data Storage and Exchange Facility, which details the development of the facility and the exchange format, describes the procedures for archiving and gives examples of the capability and use of data retrieval, is now in press at the Journal of Geophysical Research and should be published this year.

Rose Archive Bulletin
Volume 3 Number 1
06/30/82

I N D E X

SHOTLINE NUMBER PAGE NUMBER

PHASE 2

PHASE 1 EVENTS

SHOTLINE DESIGNATION: SLN1AG

Event #s Included	Num Evnts	Instrument I.D. (Origin)	Beginning and Ending Data Start Times
7184 to 7554	351	1 WHOI	1979 1/31 3:37 to 1/31 8:48
6900 to 7491	548	2 WHOI	1979 1/30 23:38 to 1/31 7:55
6286 to 6730	436	5 WHOI	1979 1/30 15: 1 to 1/30 21:15
6067 to 6495	423	6 WHOI	1979 1/30 11:57 to 1/30 17:57
6001 to 6229	222	7 WHOI	1979 1/30 11: 2 to 1/30 14:13
6001 to 6249	248	8 WHOI	1979 1/30 11: 2 to 1/30 14:30
		Subtotal	2228

SHOTLINE DESIGNATION: SLN1L

Event #s Included	Num Evnts	Instrument I.D. (Origin)	Beginning and Ending Data Start Times
5006 to 5035	29	1 WHOI	1979 2/ 3 6: 0 to 2/ 3 19:30
5006 to 5035	28	2 WHOI	1979 2/ 3 6: 0 to 2/ 3 19:30
5011 to 5035	25	6 WHOI	1979 2/ 3 8:30 to 2/ 3 19:30
5020 to 5035	16	7 WHOI	1979 2/ 3 11:59 to 2/ 3 19:30
5018 to 5035	17	8 WHOI	1979 2/ 3 11: 0 to 2/ 3 19:30
5018 to 5035	18	204 UTMSI	1979 2/ 3 10:59 to 2/ 3 19:29
5016 to 5035	19	205 UTMSI	1979 2/ 3 9:59 to 2/ 3 19:29
5007 to 5035	26	207 UTMSI	1979 2/ 3 6:29 to 2/ 3 19:29
5010 to 5013	24	208 UTMSI	1979 2/ 3 7:59 to 2/ 4 8:59
5016 to 5035	20	209 UTMSI	1979 2/ 3 9:59 to 2/ 3 19:29
5016 to 5035	18	210 UTMSI	1979 2/ 3 9:59 to 2/ 3 19:29
5006 to 5026	13	211 UTMSI	1979 2/ 3 5:59 to 2/ 3 14:59
5006 to 5026	15	212 UTMSI	1979 2/ 3 6: 0 to 2/ 3 14:59
5006 to 5026	18	213 UTMSI	1979 2/ 3 5:59 to 2/ 3 14:59
5006 to 5029	20	302 SCRIPPS	1979 2/ 3 5:59 to 2/ 3 16:29
5006 to 5035	28	303 SCRIPPS	1979 2/ 3 5:59 to 2/ 3 19:29
5006 to 5035	29	402 UW	1979 2/ 3 6: 0 to 2/ 3 19:30
5006 to 5035	29	403 UW	1979 2/ 3 6: 0 to 2/ 3 19:30
5006 to 5035	28	405 UW	1979 2/ 3 6: 0 to 2/ 3 19:30
5006 to 5035	28	407 UW	1979 2/ 3 6: 0 to 2/ 3 19:30
5020 to 5035	16	501 HIG	1979 2/ 3 11:59 to 2/ 3 19:30

Event #s Included	Num Evnts	Instrument I.D. (Origin)	Beginning and Ending Data Start Times
5006 to 5035	29	506 HIG	1979 2/ 3 6: 0 to 2/ 3 19:30
5006 to 5035	29	509 HIG	1979 2/ 3 6: 0 to 2/ 3 19:30
5006 to 5035	29	510 HIG	1979 2/ 3 6: 0 to 2/ 3 19:30
5010 to 5016	7	701 MIT	1979 2/ 3 8: 0 to 2/ 3 10: 0
5006 to 5022	32	1001 MIT	1979 2/ 3 6: 0 to 2/ 3 12:59
5023 to 5035	13	1001 MIT	1979 2/ 3 13:29 to 2/ 3 19:30
		Subtotal	603

SHOTLINE DESIGNATION: SLN1S

Event #s Included	Num Evnts	Instrument I.D. (Origin)	Beginning and Ending Data Start Times
1049 to 1167	104	2 WHOI	1979 2/ 5 1:35 to 2/ 5 11:25
1168 to 1295	113	2 WHOI	1979 2/ 5 11:30 to 2/ 5 22: 5
1175 to 1377	178	5 WHOI	1979 2/ 5 12: 5 to 2/ 6 4:55
1216 to 1377	146	6 WHOI	1979 2/ 5 15:30 to 2/ 6 4:55
1263 to 1377	99	7 WHOI	1979 2/ 5 19:25 to 2/ 6 4:55
1253 to 1377	107	8 WHOI	1979 2/ 5 18:35 to 2/ 6 4:55
1084 to 1324	20	205 UTMSI	1979 2/ 5 4:29 to 2/ 6 0:29
1144 to 1318	37	207 UTMSI	1979 2/ 5 9:29 to 2/ 6 23:59
1042 to 1162	13	211 UTMSI	1979 2/ 5 0:59 to 2/ 5 10:59
1042 to 1144	18	212 UTMSI	1979 2/ 5 0:59 to 2/ 5 9:29
1042 to 1054	66	302 SCRIPPS	1979 2/ 5 0:59 to 3/21 0:43
1042 to 1342	72	303 SCRIPPS	1979 2/ 5 0:59 to 2/ 6 1:59
1042 to 1202	160	402 UW	1979 2/ 5 1: 0 to 2/ 5 14:20
1203 to 1284	80	402 UW	1979 2/ 5 14:25 to 2/ 5 21:10
1042 to 1202	160	403 UW	1979 2/ 5 1: 0 to 2/ 5 14:20
1203 to 1284	80	403 UW	1979 2/ 5 14:25 to 2/ 5 21:10
1042 to 1202	160	405 UW	1979 2/ 5 1: 0 to 2/ 5 14:20
1203 to 1284	80	405 UW	1979 2/ 5 14:25 to 2/ 5 21:10
1042 to 1202	139	406 UW	1979 2/ 5 1: 0 to 2/ 5 14:20
1203 to 1284	68	406 UW	1979 2/ 5 14:25 to 2/ 5 21:10
1042 to 1202	160	407 UW	1979 2/ 5 1: 0 to 2/ 5 14:20
1207 to 1287	80	407 UW	1979 2/ 5 14:45 to 2/ 5 21:25
1042 to 1378	335	501 HIG	1979 2/ 5 0:59 to 2/ 6 4:59
1090 to 1378	287	506 HIG	1979 2/ 5 5: 0 to 2/ 6 4:59
1042 to 1376	329	507 HIG	1979 2/ 5 0:59 to 2/ 6 4:49
1081 to 1150	70	509 HIG	1979 2/ 5 4:15 to 2/ 5 10: 0
1153 to 1262	110	509 HIG	1979 2/ 5 10:15 to 2/ 5 19:20
1263 to 1378	115	509 HIG	1979 2/ 5 19:24 to 2/ 6 5: 0
1043 to 1243	200	510 HIG	1979 2/ 5 1: 4 to 2/ 5 17:44
1244 to 1378	134	510 HIG	1979 2/ 5 19:50 to 2/ 6 5:00
		Subtotal	3720

SHOTLINE DESIGNATION: SLN1T1

Event #s Included	Num Evnts	Instrument I.D. (Origin)	Beginning and Ending Data Start Times
3011 to 3012	2	1 WHOI	1979 2/ 2 19: 0 to 2/ 2 21: 0
3011 to 3012	2	2 WHOI	1979 2/ 2 19: 0 to 2/ 2 21: 0
4099 to 4100	2	6 WHOI	1979 2/ 2 2: 0 to 2/ 2 4: 3
4099 to 4100	2	7 WHOI	1979 2/ 2 2: 0 to 2/ 2 4: 3
4099 to 4100	2	8 WHOI	1979 2/ 2 2: 0 to 2/ 2 4: 3
4098 to 4099	2	207 UTMSI	1979 2/ 1 23:59 to 2/ 2 1:59
3011 to 3013	3	211 UTMSI	1979 2/ 2 18:59 to 2/ 2 22:59
3011 to 3013	3	212 UTMSI	1979 2/ 2 18:59 to 2/ 2 22:59
4098 to 4151	6	303 SCRIPPS	1979 2/ 2 1:59 to 2/ 6 5:59
4098 to 3013	6	402 UW	1979 2/ 2 0: 0 to 2/ 2 22:59
4098 to 3013	6	403 UW	1979 2/ 2 0: 0 to 2/ 2 22:59
4098 to 3013	6	405 UW	1979 2/ 2 0: 0 to 2/ 2 22:59
4098 to 3013	6	407 UW	1979 2/ 2 0: 0 to 2/ 2 22:59
4098 to 3013	6	502 HIG	1979 2/ 1 23:59 to 2/ 2 22:59
4098 to 3013	6	503 HIG	1979 2/ 1 23:59 to 2/ 2 23:01
4098 to 3013	6	504 HIG	1979 2/ 1 23:59 to 2/ 2 22:59
4098 to 3013	6	506 HIG	1979 2/ 1 23:59 to 2/ 2 22:59
4099 to 3013	4	507 HIG	1979 2/ 2 1:59 to 2/ 2 22:59
4099 to 3013	5	509 HIG	1979 2/ 2 1:59 to 2/ 2 22:59
4099 to 3013	5	510 HIG	1979 2/ 2 1:59 to 2/ 2 22:59
4098 to 3013	6	512 HIG	1979 2/ 1 23:59 to 2/ 2 22:59
4098 to 3013	6	514 HIG	1979 2/ 1 23:59 to 2/ 2 22:59
4099 to 4099	1	604 OSU	1979 2/ 2 2: 0 to 2/ 2 2: 0
4099 to 4100	2	1001 MIT	1979 2/ 2 2: 0 to 2/ 2 4: 2
4102 to 3013	5	1001 MIT	1979 2/ 2 13:59 to 2/ 2 22:59
		Subtotal	108

SHOTLINE DESIGNATION: SLN1T2

Event #s Included	Num Evnts	Instrument I.D. (Origin)	Beginning and Ending Data Start Times
4101 to 1005	4	6 WHOI	1979 2/ 2 12: 0 to 2/ 2 20: 0
4101 to 1005	4	7 WHOI	1979 2/ 2 12: 0 to 2/ 2 20: 0
4101 to 1005	4	8 WHOI	1979 2/ 2 12: 0 to 2/ 2 20: 0
4101 to 5005	3	204 UTMSI	1979 2/ 2 11:59 to 2/ 2 16:59
4101 to 1004	3	207 UTMSI	1979 2/ 2 11:59 to 2/ 2 17:59
1004 to 1006	3	208 UTMSI	1979 2/ 2 17:59 to 2/ 2 21:59
1004 to 1004	1	209 UTMSI	1979 2/ 2 17:59 to 2/ 2 17:59
1004 to 1006	3	210 UTMSI	1979 2/ 2 17:59 to 2/ 2 22: 0
4101 to 3013	9	302 SCRIPPS	1979 2/ 2 11:59 to 2/ 2 22:59
4101 to 1006	6	303 SCRIPPS	1979 2/ 2 11:59 to 2/ 2 21:59

4101	to	4103	3	402	UW	1979	2/	2	12:	0	to	2/	2	16: 0
1004	to	1006	3	402	UW	1979	2/	2	17:	59	to	2/	2	21:59
4101	to	4103	3	403	UW	1979	2/	2	12:	0	to	2/	2	16: 0
1004	to	1006	3	403	UW	1979	2/	2	17:	59	to	2/	2	21:59
4101	to	4103	3	405	UW	1979	2/	2	12:	0	to	2/	2	16: 0
1004	to	1006	3	405	UW	1979	2/	2	17:	59	to	2/	2	21:59
1004	to	1006	3	406	UW	1979	2/	2	17:	59	to	2/	2	21:59
4101	to	4103	3	407	UW	1979	2/	2	12:	0	to	2/	2	16: 0
1004	to	1006	3	407	UW	1979	2/	2	17:	59	to	2/	2	21:59
1004	to	1006	3	408	UW	1979	2/	2	17:	59	to	2/	2	21:59
4101	to	1006	6	502	HIG	1979	2/	2	11:	59	to	2/	2	21:59
4101	to	4103	3	503	HIG	1979	2/	2	11:	59	to	2/	2	15:59
4101	to	1006	6	504	HIG	1979	2/	2	11:	59	to	2/	2	21:59
4101	to	1006	6	506	HIG	1979	2/	2	11:	59	to	2/	2	21:59
4101	to	1006	6	507	HIG	1979	2/	2	11:	59	to	2/	2	21:59
4101	to	1006	6	509	HIG	1979	2/	2	11:	59	to	2/	2	21:59
4101	to	1006	6	510	HIG	1979	2/	2	11:	59	to	2/	2	21:59
4101	to	1006	6	512	HIG	1979	2/	2	11:	59	to	2/	2	21:59
4101	to	1006	6	514	HIG	1979	2/	2	11:	59	to	2/	2	21:59
4101	to	3013	7	603	OSU	1979	2/	2	12:	0	to	2/	2	22:59
4101	to	3013	7	604	OSU	1979	2/	2	12:	0	to	2/	2	22:59
4101	to	4101	1	1001	MIT	1979	2/	2	11:	59	to	2/	2	11:59
1004	to	1006	3	1001	MIT	1979	2/	2	17:	59	to	2/	2	21:59

Subtotal 139

SHOTLINE DESIGNATION: SLN1T3

Event #s Included			Num Evnts	Instrument I.D. (Origin)		Beginning and Ending Data Start Times								
5001	to	1003	3	1	WHOI	1979	2/	2	2:59	to	2/	2	10: 0	
5001	to	1003	4	2	WHOI	1979	2/	2	2:59	to	2/	2	10: 0	
5001	to	1003	4	6	WHOI	1979	2/	2	2:59	to	2/	2	10: 0	
5001	to	1003	3	7	WHOI	1979	2/	2	2:59	to	2/	2	10: 0	
5001	to	1003	4	8	WHOI	1979	2/	2	3: 0	to	2/	2	10: 0	
5002	to	1003	3	204	UTMSI	1979	2/	2	4:59	to	2/	2	9:59	
1001	to	1003	3	207	UTMSI	1979	2/	2	5:59	to	2/	2	9:59	
1001	to	1003	3	208	UTMSI	1979	2/	2	5:59	to	2/	2	9:59	
1001	to	1003	3	209	UTMSI	1979	2/	2	5:59	to	2/	2	9:59	
1001	to	1003	3	210	UTMSI	1979	2/	2	5:59	to	2/	2	9:59	
1001	to	1001	1	211	UTMSI	1979	2/	2	5:59	to	2/	2	5:59	
5001	to	5002	2	212	UTMSI	1979	2/	2	2:59	to	2/	2	4:59	
5001	to	1003	5	302	SCRIPPS	1979	2/	2	2:59	to	2/	2	10: 2	
5001	to	1003	5	303	SCRIPPS	1979	2/	2	2:59	to	2/	2	9:59	
5001	to	5002	2	402	UW	1979	2/	2	2:59	to	2/	2	4:59	
1001	to	1003	3	402	UW	1979	2/	2	5:59	to	2/	2	10: 0	
5001	to	5002	2	403	UW	1979	2/	2	2:59	to	2/	2	4:59	

Event #s Included		Num Evnts	Instrument I.D. (Origin)	Beginning and Ending Data Start Times
1001 to	1003	3	403 UW	1979 2/ 2 5:59 to 2/ 2 10: 0
5001 to	5002	2	405 UW	1979 2/ 2 2:59 to 2/ 2 4:59
1001 to	1003	3	405 UW	1979 2/ 2 5:59 to 2/ 2 10: 0
1001 to	1003	3	406 UW	1979 2/ 2 5:59 to 2/ 2 10: 0
5001 to	5002	2	407 UW	1979 2/ 2 2:59 to 2/ 2 4:59
1001 to	1003	3	407 UW	1979 2/ 2 5:59 to 2/ 2 10: 0
1001 to	1003	3	408 UW	1979 2/ 2 5:59 to 2/ 2 10: 0
5001 to	1003	5	502 HIG	1979 2/ 2 2:59 to 2/ 2 9:59
5001 to	5001	1	503 HIG	1979 2/ 2 2:59 to 2/ 2 2:59
5001 to	1003	5	504 HIG	1979 2/ 2 2:59 to 2/ 2 9:59
5001 to	1003	5	506 HIG	1979 2/ 2 2:59 to 2/ 2 9:59
5001 to	1003	5	507 HIG	1979 2/ 2 2:59 to 2/ 2 9:59
5001 to	1003	5	509 HIG	1979 2/ 2 2:59 to 2/ 2 9:59
5001 to	1003	5	510 HIG	1979 2/ 2 2:59 to 2/ 2 9:59
5001 to	1003	5	512 HIG	1979 2/ 2 3: 0 to 2/ 2 9:59
5001 to	1003	5	514 HIG	1979 2/ 2 2:59 to 2/ 2 9:59
1001 to	1003	3	603 OSU	1979 2/ 2 5:59 to 2/ 2 9:59
1001 to	1002	2	604 OSU	1979 2/ 2 5:59 to 2/ 2 8: 0
5001 to	1003	5	1001 MIT	1979 2/ 2 2:59 to 2/ 2 9:59
				Subtotal 123

SHOTLINE DESIGNATION: SLN1T4

Event #s Included		Num Evnts	Instrument I.D. (Origin)	Beginning and Ending Data Start Times
5004 to	5004	1	1 WHOI	1979 2/ 2 15: 0 to 2/ 2 15: 0
5004 to	5004	1	2 WHOI	1979 2/ 2 15: 0 to 2/ 2 15: 0
5003 to	5005	3	212 UTMSI	1979 2/ 2 12:59 to 2/ 2 17: 0
5003 to	5005	3	302 SCRIPPS	1979 2/ 2 12:59 to 2/ 2 16:59
5003 to	5005	3	303 SCRIPPS	1979 2/ 2 12:59 to 2/ 2 16:59
5003 to	5005	3	402 UW	1979 2/ 2 12:59 to 2/ 2 16:59
5003 to	5005	3	403 UW	1979 2/ 2 12:59 to 2/ 2 16:59
5003 to	5005	3	405 UW	1979 2/ 2 12:59 to 2/ 2 16:59
5003 to	5005	3	407 UW	1979 2/ 2 12:59 to 2/ 2 16:59
5003 to	5005	3	502 HIG	1979 2/ 2 12:59 to 2/ 2 16:59
5005 to	5005	1	503 HIG	1979 2/ 2 16:59 to 2/ 2 16:59
5003 to	5005	3	504 HIG	1979 2/ 2 12:59 to 2/ 2 16:59
5003 to	5005	3	506 HIG	1979 2/ 2 12:59 to 2/ 2 16:59
5003 to	5005	3	507 HIG	1979 2/ 2 12:59 to 2/ 2 16:59
5003 to	5005	3	509 HIG	1979 2/ 2 12:59 to 2/ 2 16:59
5003 to	5005	3	510 HIG	1979 2/ 2 12:59 to 2/ 2 16:59
5003 to	5005	3	512 HIG	1979 2/ 2 12:59 to 2/ 2 16:59
5003 to	5005	3	514 HIG	1979 2/ 2 12:59 to 2/ 2 16:59
5003 to	5005	3	1001 MIT	1979 2/ 2 12:59 to 2/ 2 16:59
				Subtotal 51

SHOTLINE DESIGNATION: SLN1T5

Event #s Included	Num Evnts	Instrument I.D. (Origin)	Beginning and Ending Data Start Times
3003 to 3010	8	1 WHOI	1979 2/ 2 10:59 to 2/ 2 14:44
3003 to 3010	8	2 WHOI	1979 2/ 2 10:59 to 2/ 2 14:44
3001 to 3002	2	204 UTMSI	1979 2/ 2 6:59 to 2/ 2 8:59
3001 to 3003	3	212 UTMSI	1979 2/ 2 6:59 to 2/ 2 10:59
3001 to 3003	3	302 SCRIPPS	1979 2/ 2 6:59 to 2/ 2 10:59
3001 to 3003	3	303 SCRIPPS	1979 2/ 2 6:59 to 2/ 2 10:59
3001 to 3010	10	402 UW	1979 2/ 2 6:59 to 2/ 2 14:44
3001 to 3010	10	403 UW	1979 2/ 2 6:59 to 2/ 2 14:44
3001 to 3010	10	405 UW	1979 2/ 2 6:59 to 2/ 2 14:44
3001 to 3010	10	407 UW	1979 2/ 2 6:59 to 2/ 2 14:44
3001 to 3010	10	502 HIG	1979 2/ 2 6:59 to 2/ 2 14:44
3001 to 3010	10	504 HIG	1979 2/ 2 6:59 to 2/ 2 14:44
3001 to 3010	10	506 HIG	1979 2/ 2 6:59 to 2/ 2 14:44
3001 to 3010	10	507 HIG	1979 2/ 2 6:59 to 2/ 2 14:44
3001 to 3010	10	509 HIG	1979 2/ 2 6:59 to 2/ 2 14:44
3001 to 3010	10	510 HIG	1979 2/ 2 6:59 to 2/ 2 14:44
3001 to 3010	10	512 HIG	1979 2/ 2 7: 0 to 2/ 2 14:44
3001 to 3010	10	514 HIG	1979 2/ 2 6:59 to 2/ 2 14:44
3001 to 3010	10	603 OSU	1979 2/ 2 6:59 to 2/ 2 14:44
3001 to 3010	10	604 OSU	1979 2/ 2 6:59 to 2/ 2 14:44
3001 to 3003	3	1001 MIT	1979 2/ 2 6:59 to 2/ 2 10:59
		Subtotal	170

SHOTLINE DESIGNATION: SLN2L

Event #s Included	Num Evnts	Instrument I.D. (Origin)	Beginning and Ending Data Start Times
4110 to 4138	26	5 WHOI	1979 2/ 4 15:45 to 2/ 5 0: 0
4104 to 4138	35	6 WHOI	1979 2/ 4 13: 0 to 2/ 5 0: 0
4104 to 4138	35	7 WHOI	1979 2/ 4 13: 0 to 2/ 5 0: 0
4105 to 4138	34	8 WHOI	1979 2/ 4 13:30 to 2/ 5 0: 0
4104 to 4121	12	204 UTMSI	1979 2/ 4 12:59 to 2/ 4 18:29
4104 to 4109	6	205 UTMSI	1979 2/ 4 12:59 to 2/ 4 15:29
4104 to 4138	23	207 UTMSI	1979 2/ 4 12:59 to 2/ 4 23:59
4113 to 4138	15	208 UTMSI	1979 2/ 4 16:29 to 2/ 4 23:59
4119 to 4138	13	209 UTMSI	1979 2/ 4 17:59 to 2/ 5 23:59
4123 to 4138	11	210 UTMSI	1979 2/ 4 18:59 to 2/ 5 23:59
4104 to 4109	6	402 UW	1979 2/ 4 13: 0 to 2/ 4 15:30
4104 to 4109	6	403 UW	1979 2/ 4 13: 0 to 2/ 4 15:30
4104 to 4109	6	405 UW	1979 2/ 4 13: 0 to 2/ 4 15:30
4104 to 4109	6	407 UW	1979 2/ 4 13: 0 to 2/ 4 15:30

4104	to	4138	35	507	HIG	1979	2/	4	12:59	to	2/ 4 23:59
4104	to	4138	35	509	HIG	1979	2/	4	12:59	to	2/ 4 23:59
4104	to	4138	35	510	HIG	1979	2/	4	12:59	to	2/ 4 23:59
4104	to	4138	35	512	HIG	1979	2/	4	13:28	to	2/ 4 23:59
4104	to	4137	34	514	HIG	1979	2/	4	12:59	to	2/ 4 23:29
4106	to	4136	30	603	OSU	1979	2/	4	13:59	to	2/ 4 22:59
4107	to	4133	21	604	OSU	1979	2/	4	14:30	to	2/ 4 21:30
4104	to	4117	14	1001	MIT	1979	2/	4	12:59	to	2/ 4 17:29
4118	to	4137	19	1001	MIT	1979	2/	4	17:44	to	2/ 4 23:29

Subtotal 492

SHOTLINE DESIGNATION: SLN2S

Event #s Included			Num Evnts	Instrument I.D. (Origin)		Beginning and Ending Data Start Times					
4140	to	4372	220	7	WHOI	1979	2/	6	5: 5	to	2/ 7 0:25
4141	to	4372	222	8	WHOI	1979	2/	6	5:10	to	2/ 7 0:25
4355	to	4403	2	205	UTMSI	1979	2/	6	22:59	to	2/ 7 2:59
4163	to	4337	23	207	UTMSI	1979	2/	6	6:59	to	2/ 6 21:29
4307	to	4409	17	210	UTMSI	1979	2/	6	18:59	to	2/ 7 3:29
4139	to	4426	276	507	HIG	1979	2/	6	4:59	to	2/ 7 4:54
4139	to	4338	191	509	HIG	1979	2/	6	4:59	to	2/ 6 21:34
4339	to	4403	62	509	HIG	1979	2/	6	21:39	to	2/ 7 02:59
4157	to	4334	170	510	HIG	1979	2/	6	6:29	to	2/ 6 21:14
4139	to	4393	208	603	OSU	1979	2/	6	4:59	to	2/ 7 2: 9
4394	to	4426	23	603	OSU	1979	2/	7	2:14	to	2/ 7 4:54
4139	to	4468	180	604	OSU	1979	2/	6	5: 0	to	2/13 2:30
4247	to	4272	22	1001	MIT	1979	2/	6	13:59	to	2/ 6 16: 4
4273	to	4290	18	1001	MIT	1979	2/	6	16: 9	to	2/ 6 17:34

Subtotal 1634

SHOTLINE DESIGNATION: SLN3A

Event #s Included			Num Evnts	Instrument I.D. (Origin)		Beginning and Ending Data Start Times					
3014	to	3122	10	208	UTMSI	1979	2/	8	11:59	to	2/ 8 16:29
3014	to	3076	62	504	HIG	1979	2/	8	11:59	to	2/ 8 14:34
3014	to	3133	119	505	HIG	1979	2/	8	11:59	to	2/ 8 16:57
3014	to	3133	119	506	HIG	1979	2/	8	11:59	to	2/ 8 16:57

Subtotal 310

SHOTLINE DESIGNATION: SLN3L

Event #s Included	Num Evnts	Instrument I.D. (Origin)	Beginning and Ending Data Start Times
1007 to 1041	35	1 WHOI	1979 2/ 3 21: 0 to 2/ 4 8: 0
1007 to 1041	34	2 WHOI	1979 2/ 3 21: 0 to 2/ 4 7:59
1007 to 1041	34	6 WHOI	1979 2/ 3 21: 0 to 2/ 4 7:59
1007 to 1041	34	7 WHOI	1979 2/ 3 21: 0 to 2/ 4 7:59
1007 to 1041	34	8 WHOI	1979 2/ 3 21: 0 to 2/ 4 7:59
1007 to 1022	11	204 UTMSI	1979 2/ 3 20:59 to 2/ 4 1:59
1007 to 1040	11	205 UTMSI	1979 2/ 3 20:59 to 2/ 4 7:29
1007 to 1041	23	207 UTMSI	1979 2/ 3 20:59 to 2/ 4 7:59
1018 to 1041	23	208 UTMSI	1979 2/ 3 0:59 to 2/ 4 7:59
1010 to 1041	20	209 UTMSI	1979 2/ 3 22:29 to 2/ 4 7:59
1011 to 1041	18	210 UTMSI	1979 2/ 3 22:59 to 2/ 4 7:59
1007 to 1041	35	402 UW	1979 2/ 3 21: 0 to 2/ 4 7:59
1007 to 1041	35	403 UW	1979 2/ 3 21: 0 to 2/ 4 7:59
1007 to 1041	35	405 UW	1979 2/ 3 21: 0 to 2/ 4 7:59
1007 to 1041	33	406 UW	1979 2/ 3 21: 0 to 2/ 4 7:59
1007 to 1041	35	407 UW	1979 2/ 3 21: 0 to 2/ 4 7:59
1007 to 1041	35	408 UW	1979 2/ 3 21: 0 to 2/ 4 7:59
1007 to 1041	35	504 HIG	1979 2/ 3 20:59 to 2/ 4 7:59
1007 to 1041	35	505 HIG	1979 2/ 3 20:59 to 2/ 4 7:59
1007 to 1041	35	506 HIG	1979 2/ 3 20:59 to 2/ 4 7:59
1007 to 1041	32	603 OSU	1979 2/ 2 21: 0 to 2/ 3 7:59
1007 to 1041	31	604 OSU	1979 2/ 2 21: 0 to 2/ 3 7:59
1007 to 1018	12	1001 MIT	1979 2/ 3 20:59 to 2/ 4 1: 0
1019 to 1030	12	1001 MIT	1979 2/ 4 1:15 to 2/ 4 4: 0
1031 to 1041	10	1001 MIT	1979 2/ 4 4:14 to 2/ 4 7:59
			Subtotal 687

SHOTLINE DESIGNATION: SLN3S

Event #s Included	Num Evnts	Instrument I.D. (Origin)	Beginning and Ending Data Start Times
1380 to 1447	68	205 UTMSI	1979 2/ 7 7: 4 to 2/ 7 12:39
1380 to 1447	68	207 UTMSI	1979 2/ 7 7: 4 to 2/ 7 12:39
1438 to 1619	160	208 UTMSI	1979 2/ 7 11:54 to 2/ 8 2:59
1584 to 1619	133	209 UTMSI	1979 2/ 7 0: 4 to 2/ 8 2:59
1477 to 1563	135	210 UTMSI	1979 2/ 7 15: 9 to 2/ 8 22:19
1480 to 1619	139	408 UW	1979 2/ 7 15:25 to 2/ 8 3: 0
1380 to 1619	240	504 HIG	1979 2/ 7 7: 4 to 2/ 8 2:59
1380 to 1619	239	506 HIG	1979 2/ 7 7: 4 to 2/ 8 2:59
			Subtotal 1182

SHOTLINE DESIGNATION: SLN4L

Event #s Included	Num Evnts	Instrument I.D. (Origin)	Beginning and Ending Data Start Times
4439 to 4473	34	1 WHOI	1979 2/12 18:00 to 2/13 5: 0
4439 to 4466	25	2 WHOI	1979 2/12 18:00 to 2/13 2: 0
4439 to 4473	33	3 WHOI	1979 2/12 18:00 to 2/13 5: 0
4439 to 4473	34	4 WHOI	1979 2/12 18:00 to 2/13 5: 0
4439 to 4473	34	7 WHOI	1979 2/12 18:00 to 2/13 5: 0
4439 to 4473	33	8 WHOI	1979 2/12 18: 0 to 2/13 5: 0
4439 to 4473	34	402 UW	1979 2/12 18: 0 to 2/13 5: 0
4439 to 4473	34	403 UW	1979 2/12 18: 0 to 2/13 5: 0
4439 to 4466	28	405 UW	1979 2/12 18: 0 to 2/13 2: 0
4439 to 4473	34	407 UW	1979 2/12 18: 0 to 2/13 5: 0
4439 to 4473	34	502 HIG	1979 2/12 17:59 to 2/13 4:59
4439 to 4473	34	503 HIG	1979 2/12 17:59 to 2/13 4:59
4468 to 4473	5	701 MIT	1979 2/13 2:30 to 2/13 5: 0
			Subtotal 396

SHOTLINE DESIGNATION: SLN4S

Event #s Included	Num Evnts	Instrument I.D. (Origin)	Beginning and Ending Data Start Times
2040 to 2252	181	1 WHOI	1979 2/11 22:35 to 2/12 16:15
2036 to 2240	181	3 WHOI	1979 2/11 22:14 to 2/12 15:15
1991 to 2224	201	4 WHOI	1979 2/11 18:30 to 2/12 13:55
2052 to 2256	173	7 WHOI	1979 2/11 23:35 to 2/12 16:35
2077 to 2263	155	8 WHOI	1979 2/12 1:39 to 2/12 17:10
2099 to 2189	13	211 UTMSI	1979 2/12 3:29 to 2/12 11: 0
1998 to 2263	277	402 UW	1979 2/11 18: 4 to 2/12 17:10
1998 to 2263	277	403 UW	1979 2/11 18: 4 to 2/12 17:10
1998 to 2263	276	405 UW	1979 2/11 18: 4 to 2/12 17:10
1998 to 2263	257	406 UW	1979 2/11 18: 4 to 2/12 17:10
1998 to 2263	277	407 UW	1979 2/11 18: 4 to 2/12 17:10
2122 to 2263	142	502 HIG	1979 2/12 5:24 to 2/12 17: 9
1998 to 2263	278	503 HIG	1979 2/11 18: 4 to 2/12 17: 9
2211 to 2263	27	701 MIT	1979 2/12 12:50 to 2/12 17:10
			Subtotal 2715

SHOTLINE DESIGNATION: SLN5L

Event #s Included	Num Evnts	Instrument I.D. (Origin)	Beginning and Ending Data Start Times
5037 to 5054	17	7 WHOI	1979 2/10 6:30 to 2/10 11:30
5048 to 5054	7	8 WHOI	1979 2/10 9:45 to 2/10 11:30
5036 to 5054	11	211 UTMSI	1979 2/10 5:59 to 2/10 11:29
5036 to 5054	11	212 UTMSI	1979 2/10 5:59 to 2/10 11:29
5036 to 5054	18	402 UW	1979 2/10 5:59 to 2/10 11:30
5036 to 5054	18	403 UW	1979 2/10 5:59 to 2/10 11:30
5036 to 5054	18	405 UW	1979 2/10 5:59 to 2/10 11:30
5036 to 5054	18	407 UW	1979 2/10 5:59 to 2/10 11:30
5036 to 5038	3	701 MIT	1979 2/10 6: 0 to 2/10 7: 0
		Subtotal	121

SHOTLINE DESIGNATION: SLN5S

Event #s Included	Num Evnts	Instrument I.D. (Origin)	Beginning and Ending Data Start Times
1620 to 1760	124	211 UTMSI	1979 2/ 8 17: 0 to 2/ 9 4:39
1654 to 1810	156	212 UTMSI	1979 2/ 8 19:49 to 2/ 9 8:49
1620 to 1810	191	402 UW	1979 2/ 8 16:59 to 2/ 9 8:50
1620 to 1810	191	403 UW	1979 2/ 8 16:59 to 2/ 9 8:50
1620 to 1810	191	405 UW	1979 2/ 8 16:59 to 2/ 9 8:50
1620 to 1810	168	406 UW	1979 2/ 8 16:59 to 2/ 9 8:50
1620 to 1810	191	407 UW	1979 2/ 8 16:59 to 2/ 9 8:50
		Subtotal	1212

SHOTLINE DESIGNATION: SLN6L

Event #s Included	Num Evnts	Instrument I.D. (Origin)	Beginning and Ending Data Start Times
4427 to 4438	12	2 WHOI	1979 2/ 8 6:59 to 2/ 8 11:30
4427 to 4431	5	208 UTMSI	1979 2/ 8 6:59 to 2/ 8 8:59
4427 to 4438	10	211 UTMSI	1979 2/ 8 6:59 to 2/ 8 11:29
4427 to 4438	12	402 UW	1979 2/ 8 6:59 to 2/ 8 11:30
4427 to 4438	12	403 UW	1979 2/ 8 6:59 to 2/ 8 11:30

4427	to	4438	12	405	UW	1979	2/	8	6:59	to	2/ 8 11:30
4427	to	4438	12	407	UW	1979	2/	8	6:59	to	2/ 8 11:30
4427	to	4438	12	503	HIG	1979	2/	8	6:59	to	2/ 8 11:29
4428	to	4437	5	603	OSU	1979	2/	8	7:29	to	2/ 8 10:59
4427	to	4438	12	701	MIT	1979	2/	8	6:59	to	2/ 8 11:30
									Subtotal		104

SHOTLINE DESIGNATION: SLN6S

Event #s Included			Num Evnts	Instrument I.D. (Origin)		Beginning and Ending Data Start Times					
3146	to	3236	16	211	UTMSI	1979	2/10	17:59	to	2/11	1:29
3134	to	3241	108	402	UW	1979	2/10	16:59	to	2/11	1:55
3134	to	3241	108	403	UW	1979	2/10	16:59	to	2/11	1:55
3134	to	3241	108	405	UW	1979	2/10	16:59	to	2/11	1:55
3134	to	3241	108	407	UW	1979	2/10	16:59	to	2/11	1:55
3134	to	3241	108	503	HIG	1979	2/10	16:59	to	2/11	1:54
3184	to	3240	38	701	MIT	1979	2/10	21:10	to	2/11	1:49
									Subtotal		594

SHOTLINE DESIGNATION: SLN7L

Event #s Included			Num Evnts	Instrument I.D. (Origin)		Beginning and Ending Data Start Times					
5057	to	5066	10	7	WHOI	1979	2/10	13: 0	to	2/10	16:30
5057	to	5066	10	8	WHOI	1979	2/10	13: 0	to	2/10	16:30
5057	to	5065	6	211	UTMSI	1979	2/10	12:59	to	2/10	15:59
5057	to	5065	10	212	UTMSI	1979	2/10	12:59	to	2/11	15:59
5057	to	5068	12	402	UW	1979	2/10	13: 0	to	2/10	17:30
5057	to	5068	12	403	UW	1979	2/10	13: 0	to	2/10	17:30
5057	to	5068	12	405	UW	1979	2/10	13: 0	to	2/10	17:30
5057	to	5068	12	407	UW	1979	2/10	13: 0	to	2/10	17:30
									Subtotal		84

SHOTLINE DESIGNATION: SLN7S

Event #s Included	Num Evnts	Instrument I.D. (Origin)	Beginning and Ending Data Start Times
5069 to 5171	18	212 UTMSI	1979 2/11 8:59 to 2/11 17:29
5069 to 5176	108	402 UW	1979 2/11 9: 0 to 2/11 17:54
5069 to 5152	84	403 UW	1979 2/11 9: 0 to 2/11 15:55
5069 to 5176	106	405 UW	1979 2/11 9: 0 to 2/11 17:54
5069 to 5176	108	407 UW	1979 2/11 9: 0 to 2/11 17:54
			Subtotal 424

SHOTLINE DESIGNATION: QUAKE1

Event #s Included	Num Evnts	Instrument I.D. (Origin)	Beginning and Ending Data Start Times
1 to 1	1	509 HIG	1979 2/ 2 0:54 to 2/ 2 0:54
1 to 46	46	517 HIG	1979 2/27 21:30 to 3/ 8 18:33
1 to 2	2	701 MIT	1979 2/12 1:59 to 2/12 2: 0
			Subtotal 49

Total Number of Events for Phase 1: 17,164

PHASE 2 EVENTS

SHOTLINE DESIGNATION: SLN1N

Event #s Included	Num Evnts	Instrument I.D. (Origin)	Beginning and Ending Data Start Times
3242 to 3330	89	5 WHOI	1979 2/28 0: 0 to 2/28 7:19
3242 to 3330	89	6 WHOI	1979 2/28 0: 0 to 2/28 7:19
3244 to 3319	34	210 UTMSI	1979 2/28 0:10 to 2/28 6:25
3242 to 3304	63	213 UTMSI	1979 2/28 0: 0 to 2/28 5:10
3242 to 3331	88	608 OSU	1979 2/28 0: 0 to 2/28 7:25
3242 to 3331	89	611 OSU	1979 2/28 0: 0 to 2/28 7:25
3256 to 3331	71	612 OSU	1979 2/28 1:10 to 2/28 7:25
		Subtotal	523

SHOTLINE DESIGNATION: SLN2N

Event #s Included	Num Evnts	Instrument I.D. (Origin)	Beginning and Ending Data Start Times
3370 to 3421	52	205 UTMSI	1979 2/28 18:10 to 2/28 22:24
3333 to 3419	79	210 UTMSI	1979 2/28 15: 5 to 2/28 22:15
3332 to 3421	89	608 OSU	1979 2/28 15: 0 to 2/28 22:24
3332 to 3403	72	611 OSU	1979 2/28 15: 0 to 2/28 20:55
3404 to 3421	18	611 OSU	1979 2/28 20:59 to 2/28 22:24
3332 to 3421	75	612 OSU	1979 2/28 15: 0 to 2/28 22:24
		Subtotal	385

SHOTLINE DESIGNATION: SLN3N

Event #s Included	Num Evnts	Instrument I.D. (Origin)	Beginning and Ending Data Start Times
4474 to 4563	86	2 WHOI	1979 2/28 7:30 to 2/28 14:55
4474 to 4563	85	3 WHOI	1979 2/28 7:30 to 2/28 14:55
4506 to 4563	14	210 UTMSI	1979 2/28 10:10 to 2/28 14:55

4474	to	4538	64	608	OSU	1979	2/28	7:30	to 2/28 12:50
4540	to	4563	23	608	OSU	1979	2/28	13: 0	to 2/28 14:55
4474	to	4563	88	611	OSU	1979	2/28	7:30	to 2/28 14:55
4474	to	4538	63	612	OSU	1979	2/28	7:30	to 2/28 12:50
4540	to	4562	20	612	OSU	1979	2/28	13: 0	to 2/28 14:49
								Subtotal	443

SHOTLINE DESIGNATION: SLN4N

Event #s Included			Num Evnts	Instrument I.D. (Origin)		Beginning and Ending Data Start Times				
4573	to	4655	79	210	UTMSI	1979	2/28	23:15	to 3/ 1	6:54
4641	to	4655	14	213	UTMSI	1979	2/28	5:44	to 2/28	6:54
4642	to	4649	7	214	UTMSI	1979	3/ 1	5:50	to 3/ 1	6:25
4564	to	4655	87	608	OSU	1979	2/28	22:30	to 3/ 1	6:54
4564	to	4655	90	611	OSU	1979	2/28	22:30	to 3/ 1	6:54
4564	to	4655	83	612	OSU	1979	2/28	22:30	to 3/ 1	6:54
								Subtotal	360	

SHOTLINE DESIGNATION: SLN5N

Event #s Included			Num Evnts	Instrument I.D. (Origin)		Beginning and Ending Data Start Times				
3425	to	3427	3	1	WHOI	1979	3/ 4	17:20	to 3/ 4	22:15
3425	to	3427	3	2	WHOI	1979	3/ 4	17:20	to 3/ 4	22:15
3425	to	3427	3	3	WHOI	1979	3/ 4	17:20	to 3/ 4	22:15
3425	to	3427	3	4	WHOI	1979	3/ 4	17:20	to 3/ 4	22:15
3425	to	3427	3	5	WHOI	1979	3/ 4	17:20	to 3/ 4	22:15
3425	to	3427	3	6	WHOI	1979	3/ 4	17:20	to 3/ 4	22:15
3425	to	3427	3	7	WHOI	1979	3/ 4	17:20	to 3/ 4	22:15
3425	to	3427	3	8	WHOI	1979	3/ 4	17:20	to 3/ 4	22:15
3425	to	3425	1	204	UTMSI	1979	3/ 4	17:20	to 3/ 4	17:20
3425	to	3427	3	205	UTMSI	1979	3/ 4	17:20	to 3/ 4	22:15
3425	to	3427	3	210	UTMSI	1979	3/ 4	17:20	to 3/ 4	22:15
3426	to	3427	2	213	UTMSI	1979	3/ 4	20:15	to 3/ 4	22:15
3425	to	3426	2	214	UTMSI	1979	3/ 4	17:20	to 3/ 4	20:14
3425	to	3427	3	302	SCRIPPS	1979	3/ 4	17:20	to 3/ 4	22:15
3425	to	3427	3	303	SCRIPPS	1979	3/ 4	17:20	to 3/ 4	22:15
3425	to	3427	3	304	SCRIPPS	1979	3/ 4	17:21	to 3/ 4	22:16
3425	to	3427	3	608	OSU	1979	3/ 4	17:20	to 3/ 4	22:15
3425	to	3427	3	611	OSU	1979	3/ 4	17:20	to 3/ 4	22:15
3425	to	3427	2	612	OSU	1979	3/ 4	17:20	to 3/ 4	22:15

3425	to	3427	3	705	MIT	1979	3/	4	17:21	to 3/ 4 22:15
3425	to	3427	3	706	MIT	1979	3/	4	17:21	to 3/ 4 22:15
3425	to	3427	3	801	UCSB	1979	3/	4	17:21	to 3/ 4 22:15
									Subtotal	61

SHOTLINE DESIGNATION: SLN6N

Event #s Included			Num Evnts	Instrument I.D. (Origin)		Beginning and Ending Data Start Times					
3428	to	3432	5	1	WHOI	1979	3/	5	1:14	to 3/ 5	22:46
3428	to	3432	5	2	WHOI	1979	3/	5	1:14	to 3/ 5	22:46
3428	to	3432	5	3	WHOI	1979	3/	5	1:14	to 3/ 5	22:46
3428	to	3430	2	4	WHOI	1979	3/	5	1:14	to 3/ 5	15:44
3428	to	3432	5	5	WHOI	1979	3/	5	1:14	to 3/ 5	22:46
3428	to	3432	5	6	WHOI	1979	3/	5	1:14	to 3/ 5	22:46
3428	to	3432	5	7	WHOI	1979	3/	5	1:14	to 3/ 5	22:46
3428	to	3432	5	8	WHOI	1979	3/	5	1:14	to 3/ 5	22:46
3428	to	3432	5	204	UTMSI	1979	3/	5	1:14	to 3/ 5	22:46
3428	to	3432	5	205	UTMSI	1979	3/	5	1:14	to 3/ 5	22:46
3428	to	3432	5	210	UTMSI	1979	3/	5	1:14	to 3/ 5	22:46
3428	to	3432	3	213	UTMSI	1979	3/	5	1:14	to 3/ 5	22:46
3430	to	3432	3	301	SCRIPPS	1979	3/	5	15:44	to 3/ 5	22:47
3428	to	3432	5	302	SCRIPPS	1979	3/	5	1:14	to 3/ 5	22:46
3428	to	3432	5	303	SCRIPPS	1979	3/	5	1:14	to 3/ 5	22:47
3428	to	3428	1	304	SCRIPPS	1979	3/	5	1:14	to 3/ 5	1:14
3428	to	3432	5	608	OSU	1979	3/	5	1:14	to 3/ 5	22:47
3428	to	3432	5	611	OSU	1979	3/	5	1:14	to 3/ 5	22:47
3428	to	3431	3	612	OSU	1979	3/	5	1:14	to 3/ 5	19:21
3428	to	3432	5	705	MIT	1979	3/	5	1:14	to 3/ 5	22:47
3428	to	3432	5	706	MIT	1979	3/	5	1:14	to 3/ 5	22:46
3428	to	3428	1	801	UCSB	1979	3/	5	1:14	to 3/ 5	1:14
3430	to	3432	2	802	UCSB	1979	3/	5	15:45	to 3/ 5	22:47
									Subtotal		95

SHOTLINE DESIGNATION: SLN7N

Event #s Included			Num Evnts	Instrument I.D. (Origin)		Beginning and Ending Data Start Times					
3433	to	3438	6	1	WHOI	1979	3/	6	1:26	to 3/ 6	23:37
3433	to	3438	6	2	WHOI	1979	3/	6	1:26	to 3/ 6	23:37
3433	to	3438	6	3	WHOI	1979	3/	6	1:26	to 3/ 6	23:37
3433	to	3438	6	4	WHOI	1979	3/	6	1:26	to 3/ 6	23:37

3433	to	3438	6	5	WHOI	1979	3/	6	1:26	to	3/ 6 23:37
3433	to	3438	6	6	WHOI	1979	3/	6	1:26	to	3/ 6 23:37
3433	to	3438	6	7	WHOI	1979	3/	6	1:26	to	3/ 6 23:37
3433	to	3438	6	8	WHOI	1979	3/	6	1:26	to	3/ 6 23:37
3433	to	3438	6	204	UTMSI	1979	3/	6	1:26	to	3/ 6 23:37
3433	to	3438	6	205	UTMSI	1979	3/	6	1:26	to	3/ 6 23:37
3433	to	3437	4	210	UTMSI	1979	3/	6	1:26	to	3/ 6 21:37
3433	to	3438	5	213	UTMSI	1979	3/	6	1:26	to	3/ 6 23:37
3433	to	3438	6	301	SCRIPPS	1979	3/	6	1:26	to	3/ 6 23:37
3433	to	3438	5	302	SCRIPPS	1979	3/	6	1:26	to	3/ 6 23:38
3433	to	3438	6	303	SCRIPPS	1979	3/	6	1:26	to	3/ 6 23:38
3433	to	3438	6	608	OSU	1979	3/	6	1:26	to	3/ 6 23:37
3433	to	3438	6	611	OSU	1979	3/	6	1:26	to	3/ 6 23:37
3433	to	3438	6	705	MIT	1979	3/	6	1:26	to	3/ 6 23:38
3433	to	3438	6	706	MIT	1979	3/	6	1:26	to	3/ 6 23:37
3433	to	3438	6	802	UCSB	1979	3/	6	1:26	to	3/ 6 23:37

Subtotal 116

SHOTLINE DESIGNATION: SLN8N

Event #s Included			Num Evnts	Instrument I.D. (Origin)		Beginning and Ending Data Start Times					
3440	to	3442	3	1	WHOI	1979	3/	7	17:20	to	3/ 7 22:45
3440	to	3442	3	2	WHOI	1979	3/	7	17:20	to	3/ 7 22:45
3440	to	3442	3	3	WHOI	1979	3/	7	17:20	to	3/ 7 22:45
3440	to	3442	3	4	WHOI	1979	3/	7	17:20	to	3/ 7 22:45
3440	to	3442	3	5	WHOI	1979	3/	7	17:20	to	3/ 7 22:45
3440	to	3442	3	6	WHOI	1979	3/	7	17:20	to	3/ 7 22:45
3440	to	3442	3	7	WHOI	1979	3/	7	17:20	to	3/ 7 22:45
3440	to	3442	3	8	WHOI	1979	3/	7	17:20	to	3/ 7 22:45
3442	to	3442	1	204	UTMSI	1979	3/	7	22:45	to	3/ 7 22:45
3442	to	3442	1	205	UTMSI	1979	3/	6	22:45	to	3/ 6 22:45
3441	to	3442	2	210	UTMSI	1979	3/	7	20:12	to	3/ 7 20:45
3440	to	3442	3	301	SCRIPPS	1979	3/	7	17:20	to	3/ 7 22:45
3440	to	3442	3	302	SCRIPPS	1979	3/	7	17:21	to	3/ 7 22:45
3440	to	3442	3	303	SCRIPPS	1979	3/	7	17:21	to	3/ 7 22:45
3441	to	3442	2	608	OSU	1979	3/	7	20:12	to	3/ 7 22:45
3440	to	3442	3	611	OSU	1979	3/	7	17:20	to	3/ 7 22:45
3440	to	3442	3	705	MIT	1979	3/	7	17:20	to	3/ 7 22:45
3440	to	3442	3	706	MIT	1979	3/	7	17:20	to	3/ 7 22:45
3440	to	3442	3	802	UCSB	1979	3/	7	17:21	to	3/ 7 22:45

Subtotal 51

SHOTLINE DESIGNATION: QUAKE2

Event #s Included	Num Evnts	Instrument I.D. (Origin)	Beginning and Ending Data Start Times
5 to 68	39	202 UTMSI	1979 2/27 15:47 to 3/ 1 20:49
1 to 55	9	203 UTMSI	1979 2/27 13: 7 to 3/ 1 15: 6
24 to 329	141	204 UTMSI	1979 2/27 22:38 to 3/15 17:56
24 to 330	97	205 UTMSI	1979 2/27 22:38 to 3/15 18:35
24 to 332	52	210 UTMSI	1979 2/27 22:38 to 3/16 7: 0
12 to 267	59	213 UTMSI	1979 2/27 18: 8 to 3/ 8 18:32
2 to 230	143	214 UTMSI	1979 2/27 13:25 to 3/ 8 8:52
1014 to 1399	186	301 SCRIPPS	1979 3/ 1 14:57 to 3/13 4:13
2004 to 2098	16	302 SCRIPPS	1979 3/ 4 4:59 to 4/21 9:54
3004 to 3134	21	303 SCRIPPS	1979 3/ 3 14:59 to 3/ 9 17:38
4100 to 4408	64	304 SCRIPPS	1979 3/ 2 0:17 to 3/ 5 8:42
1 to 130	130	608 OSU	1979 2/27 11:30 to 3/14 14: 3
131 to 275	145	608 OSU	1979 3/14 14:16 to 3/17 4:38
1 to 121	121	611 OSU	1979 2/27 22:38 to 3/14 20:38
146 to 226	81	611 OSU	1979 3/15 5:12 to 3/20 22:57
1 to 108	108	612 OSU	1979 3/ 1 6:54 to 3/16 6:59
1 to 260	258	614 OSU **	1979 3/26 0:37 to 3/31 15:15
261 to 516	255	614 OSU **	1979 3/31 18:23 to 4/ 6 5:51
517 to 664	146	614 OSU **	1979 4/ 6 6:12 to 4/10 18: 4
1 to 138	138	616 OSU **	1979 3/26 15:28 to 4/11 22:59
0 to 0	19	705 MIT	1979 3/ 2 5:23 to 3/13 18:49
0 to 0	37	706 MIT	1979 3/ 2 5:23 to 3/ 8 12:38
0 to 0	78	801 UCSB	1979 3/ 1 12:16 to 3/ 5 0:58
0 to 0	49	802 UCSB	1979 3/ 5 8:40 to 3/ 9 0:43
			Subtotal 2392

** These events were collected after Phase II; a third deployment off the coast of Mexico, near the Petatlan site.

Total Number of Events for Phase 2: 4,426

SUMMARY OF EVENTS BY INSTITUTE

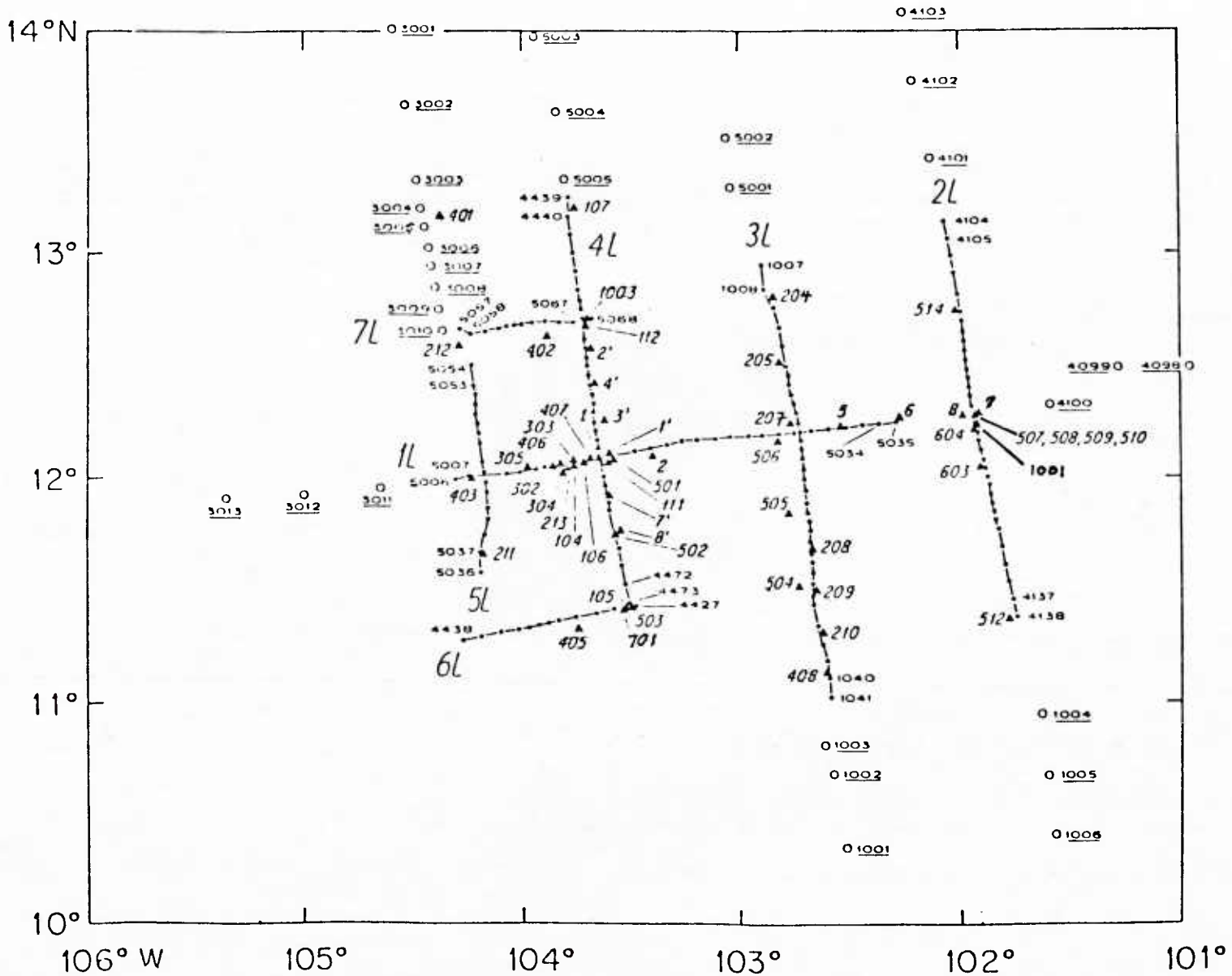
HIG	4,377
MIT	342
OSU	3,010
SCRIPPS	565
UCSB	142
UTMSI	2,424
UW	5,216
WHOI	5,514
TOTAL STORED EVENTS	21,590

Rose Shotline Number Designations used in Rose Archive

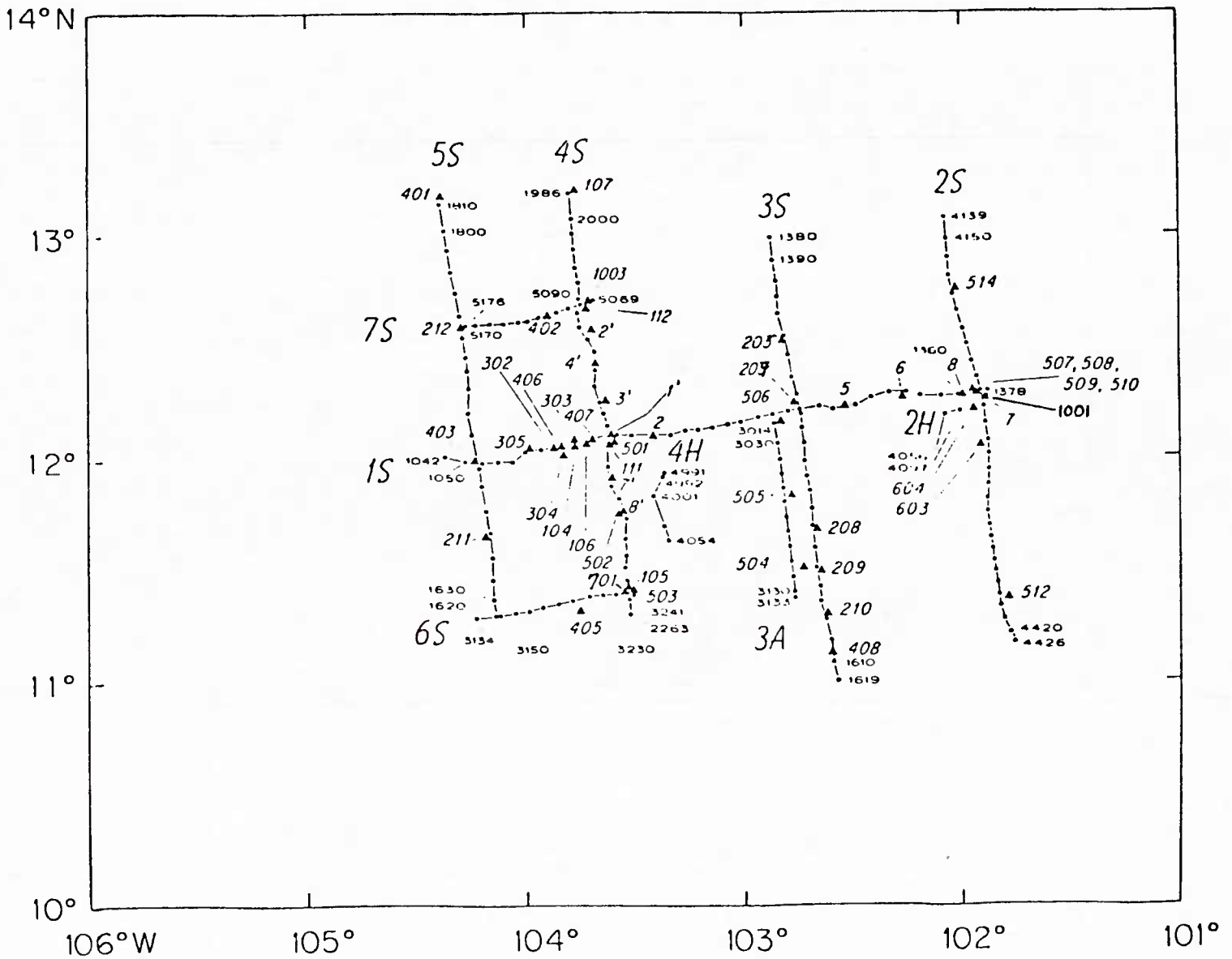
PHASE 1	Shotline	Shot numbers	Ship code
	1L	5006-5035	D
	1S	1042-1378	T
	2L	4104-4138	K
	2S	4139-4426	K
	3L	1007-1041	T
	3S	1379-1619	T
	3A	3014-3133	C
	4L	4439-4473 & 4993	K
	4S	1986-2263	T
	5L	5036-5054	D
	5S	1620-1810	T
	6L	4427-4438	K
	6S	3134-3241	C
	7L	5057-5068	D
	7S	5069-5176	D
	4H	4991-4992 & 4001-4054	K
	2H	4056-4097	K
		<< 1 TON SHOTS >>	
	1T1	3011-3013	C
		& 4098-4100	K
	1T2	1004-1006	T
		& 4101-4103	K
	1T3	1001-1003	T
		& 5001-5002	D
	1T4	5003-5005	D
	1T5	3001-3010	C
		<< MAXIPULSE AND AIRGUN >>	
	1M	8001-8722	K
	2M	9001-11360	K
	3M	11362-12219	K
	4M	12220-12639	K
	1AG	6001-7640	C

PHASE 2	Shotline	Shot numbers	Ship code
	1N	3242-3331	C
	2N	3332-3421	C
	3N	4474-4563	K
	4N	4564-4655	K
	5N	3425-3427	C
	6N	3428-3432	C
	7N	3433-3438	C
	8N	3439-3442	C
	5T	4927-4928	K

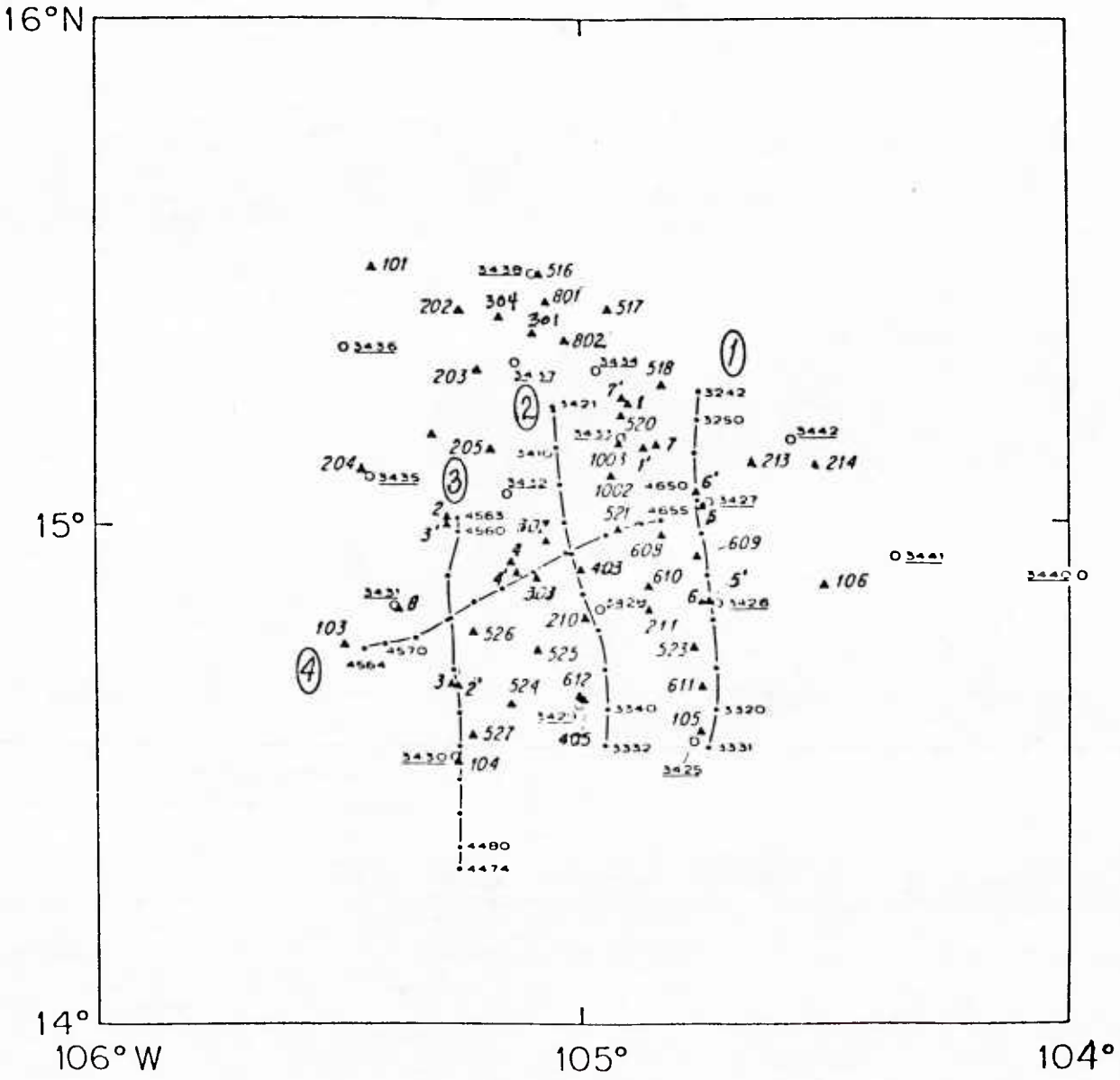
Ship codes: C=Conrad; D=DeSteiger; K=Kana Keoki; and T= Thompson



Locations of instrument deployments and large shots in Phase I. Open circles denote 800 kg. shots. Solid circles are 200 kg or 800 kg.



Locations of instrument deployments and small shots in Phase I. Small shots alternating 2 kg and 5 kg. Figure shows location of every 10th shot.



Locations of instrument deployments and shots for Phase II.
Open circles are 200 kg, except 3440 and 3441, which are 800 kg.

APPENDIX C

The H.I.G. Computing Facility

The HIG computing facility equipment is supplied by Harris Computer Systems. The hardware either installed or on order consists of:

Model H800 CPU with Virtual Memory Access
4096 Virtual Memory Paging Registers
  (i.e. 4096k bytes) of MOS memory
(3) 300 Megabyte Disc Modules
(3) Disc Controllers and Channels
(4) Tape Drives
    (3) 800/1600 BPI
    (1) 800 BPI
(2) Tape controllers
(56) Terminal Ports
(1) VERSATEC Printer/Plotter
(1) SPINWRITER Word Processing Printer
(1) Analog/Digital Controller and Channel

The system utilizes the VULCAN Operating System. VULCAN supports time sharing among the terminals concurrent with the execution of batch jobs. VULCAN also includes a number of programming languages. These include Fortran, Cobol, Pascal, APL, and Basic as well as a number of utilities for sorting and editing of both text and data.

APPENDIX D

PROGRAM LISTINGS

```
 1: C PROGRAM DISTAP TO DISPLAY CONTENTS OF ROSE ARCHIVE TAPE HEADER
 2: C
 3: C      WRITTEN BY SHARON LATRAILLE   LAST UPDATED 6/24/82
 4: C
 5:       NAME DISTAP
 6:       INTEGER THBUF(224),TRAY(7),RDATE(3),JDATE(3)
 7:       INTEGER IDATE(3),KDATE(3),DBUF(54),JBUF(260),IS(5),IE(5)
 8:       INTEGER TAPE,DEPIMN,DEPEMX,DEPEMN,NBUF(3)
 9:       INTEGER ENUMMX,ENUMMN,EXPLMX,EXPLMN,ICHNMX
10:       INTEGER ICHNMN,TYPEMX,TYPEMN,EDEPMX,EDEPMN
11:       INTEGER F(3,8),F1(3),F2(3),F3(3),F4(3),F5(3),F6(3),F7(3)
12:       INTEGER F8(3)
13:       REAL ILATMN,ILONMN
14:       DIMENSION PLACE(5),DNAME(9),EXCODE(4),SHTLN(10)
15:       INTEGER*6 CSEC,DSTMAX,DSTMIN,TDST,TDET
16:       INTEGER*1 ICRAY(162),DCBUF(162),NFBUF(9)
17:       EQUIVALENCE (JBUF(1),ICRAY(1)),(DBUF,DCBUF),(NBUF,NFBUF)
18:       EQUIVALENCE (F1,F),(F2,F(1,2)),(F3,F(1,3)),(F4,F(1,4))
19:       EQUIVALENCE (F5,F(1,5)),(F6,F(1,6)),(F7,F(1,7)),(F8,F(1,8))
20:       EQUIVALENCE (F9,F(1,9)),(F10,F(1,10)),(F11,F(1,11)),(F12,F(1,12))
21:       COMMON/ITCM/TRAY,CSEC,JULD
22:       DATA F1/'(1X,I8)'/,F2/'(1X,I7)'/,F3/'(1X,I6)'/,F4/'(1X,I5)'/
23:       DATA F5/'(1X,I4)'/,F6/'(1X,I3)'/,F7/'(1X,I2)'/,F8/'(1X,I1)'/
24:       DATA F9/'(I4)'/,F10/'(I3)'/,F11/'(I2)'/,F12/'(I1)'/
25:       DATA TAPE/4/
26: C
27:       OPEN TAPE
28:       NREC=0
29:       LOOP(1)
30:        . WRITE(3,1000)
31: 1000   . FORMAT(1X,'Enter 0 if format is that of an incoming ROSE',
32:       +. ' formatted tape, or',/ 'Enter 1 if tape begins with an HIG',
33:       +. ' Archive header:')
34:        . READ(0,)MTYPE
35:        . IF(MTYPE.EQ.0)
36:        . . JDIS=0
37:        . . WRITE(3,1002)
38: 1002   . . FORMAT(1X,'Display first 80 words of "N" headers. Enter N:',
39:       +. . /' or, optionally display 5 words of all headers. Enter 0:')
40:        . . READ(0,) NWDS
41:        . END IF
42:        . EXIT LOOP IF (MTYPE.EQ.1)
43: 25     . BUFFER IN(TAPE,JBUF,B,256,ISTAT,IL)
44:        . NREC=NREC+1
45:        . CALL STATUS(TAPE)
46:        . IF(ISTAT.EQ.3) GO TO 25
47:        . J=1
48:        . FOR I=1,162
49:        . . K=MOD(I,3)
50:        . . IF(K.NE.1)
51:        . . . DCBUF(J)=ICRAY(I)
52:        . . . J=J+1
53:        . . END IF
54:        . END FOR
55: C DECODE NUMBER OF FILES
56:        . J=1
57:        . I=0
58:        . FOR K=108,101,-1
59:        . . IF(DCBUF(K).NE.' ')
60:        . . . I=I+1
61:        . . . IF(I.EQ.1)
62:        . . . . IP=J
63:        . . . END IF
64:        . . END IF
65:        . . J=J+1
66:        . END FOR
67:        . DECODE(9,F(1,IP),DBUF(34)) NFILES
68: C
69: C DECODE IID
70:        . J=1
71:        . I=0
72:        . FOR K=4,1,-1
73:        . . IF(DCBUF(K).NE.' ')
74:        . . . I=I+1
75:        . . . IF(I.EQ.1)
76:        . . . . IP=J+8
77:        . . . END IF
78:        . . END IF
79:        . . J=J+1
80:        . END FOR
```

```
 81:       .  DECODE(4,F(1,IP),DBUF) IID
 82: C     DECODE(4,7999,DBUF)IID
 83: C7999 FORMAT(I4)
 84:       .  DECODE(55,8005,DBUF(2)) DNAME
 85: 8005  .  FORMAT(1X,9A6)
 86:       .  DECODE(20,8006,DBUF(21)) EXCODE
 87: 8006  .  FORMAT(3A6,A2)
 88:       .  DECODE(12,8010,DBUF(27)) (IS(I),I=1,5)
 89:  8010 .  FORMAT(2X,5I2)
 90:       .  DECODE(10,8020,DBUF(31)) (IE(I),I=1,5)
 91:  8020 .  FORMAT (5I2)
 92:       .  WRITE(3,8040)IID,NFILES
 93:  8040 .  FORMAT(1X,"INSTRUMENT ID  ",I4,/," # FILES  ",I4)
 94:       .  WRITE(3,8045) DNAME
 95: 8045  .  FORMAT(1X,'DESIGNER NAME & ADDRESS: ',9A6)
 96:       .  WRITE(3,8046) EXCODE
 97: 8046  .  FORMAT(1X,3A6,A2,' EXPERIMENT')
 98:       .  WRITE(3,8050) IS
 99: 8050  .  FORMAT(1X,"START TIME  ",5(1X,I2))
100:       .  WRITE(3,8060) IE
101: 8060  .  FORMAT(1X,"END TIME    ",5(1X,I2))
102:       .  WRITE(10,8040)IID,NFILES
103:       .  WRITE(10,8045) DNAME
104:       .  WRITE(10,8046) EXCODE
105:       .  WRITE(10,8050) IS
106:       .  WRITE(10,8060) IE
107: C DISPLAY 1ST 80 WORDS OF THE FIRST FILE HEADER RECORD
108: C OR DISPLAY 5 WORDS OF EACH HEADER
109:       .    IF(NWDS.EQ.0)
110:       . .    JDIS=1
111:       . .    NWDS=NFILES
112:       .    END IF
113:       .    LOOP(NWDS)
114:       . .    DO
115:       . . .    BUFFER IN (TAPE,JBUF,B,260,IFST,ILEN)
116:       . . .  NREC=NREC+1
117:       . . .    CALL STATUS(TAPE)
118:       . . .    IF(IFST.EQ.3)
119:       . . . .    WRITE(3,7602)
120: 7602  . . . .    FORMAT(1X,'END OF FILE ENCOUNTERED')
121:       . . .    END IF
122:       . . .    EXIT LOOP IF(IFST.GE.4)
123:       . .    UNTIL(ILEN.EQ.256)
124:       . .    CALL CNVNEG(JBUF,256)
125:       . .    IF(JDIS.EQ.1)
126:       . . .    WRITE(10,8301) JBUF(41),JBUF(1),JBUF(2),JBUF(3),JBUF(71)
127: 8301  . . .    FORMAT(' FILE #:',I4,'   WORDS 1,2,3,71:',4I8)
128:       . .    ELSE
129:       . . .    FOR II=1,80
130:       . . . .    WRITE(10,8300) II,JBUF(II)
131: 8300  . . . .    FORMAT(1X,'FILE HEADER WORD #',I2,' : ',I8)
132:       . . .    END FOR
133:       . .    END IF
134:       .    END LOOP
135:       .  WRITE(3,) NREC
136:       END LOOP
137:       LOOP(1)
138:       .  EXIT LOOP IF(MTYPE.EQ.0)
139:       .  BUFFER IN(TAPE,THBUF,B,224,ISTAT,IL)
140:       .  CALL STATUS(TAPE)
141:       .  DECODE(9,4000,THBUF) ITAP
142: 4000  .  FORMAT(3X,I6)
143:       .  DECODE(75,4001,THBUF(4)) NSLN,(PLACE(I),I=1,5),IDOC,
144:      +.  (RDATE(I),I=1,3),(IDATE(I),I=1,3),(JDATE(I),I=1,3),
145:      +.  (KDATE(I),I=1,3)
146: 4001  .  FORMAT(I6,5A6,13A3)
147:       .  DECODE(96,6000,THBUF(29)) INUMMN,ENUMMX,ENUMMN,DSTMAX,
148:      +.  DSTMIN,EXPLMX,EXPLMN,DEPIMN,DEPEMX,DEPEMN,
149:      +.  IDEPMN,EDEPMX,EDEPMN,ICHNMX,ICHNMN,TYPEMX,TYPEMN
150: 6000  .  FORMAT(3I6,2I13,2I2,6I6,2I3,2I2,2X)
151:       .  DECODE(102,6001,THBUF(61)) SIZEMX,SIZEMN,RANGMX,RANGMN,
152:      +.  ILATMN,ILONMN,ELATMX,ELATMN,ELONMX,ELONMN
153: 6001  .  FORMAT(10F10.4,2X)
154:       .  DECODE(174,6002,THBUF(95)) IID,DNAME,EXCODE,NFILES,TDST,TDET,SHTLN
155: 6002  .  FORMAT(I4,12A6,A2,I6,I14,1X,I14,1X,10A6)
156:       .  DECODE(3,6003,THBUF(160)) IDASH
157: 6003  .  FORMAT(I3)
158:       .  WRITE(10,2001) ITAP,IDASH
159: 2001  .  FORMAT(1X,'ROSE ARCHIVE TAPE NO.',T40,I6,'-',I3)
160:       .  IF(NSLN.EQ.1)
```

```
161:        .  .  WRITE(10,2002) SHTLN(1)
162: 2002   .  .  FORMAT(1X,'SHOT LINE NO.',T40,A6)
163:        .  END IF
164:        .  IF(NSLN.GT.1)
165:        .  .  WRITE(10,2050) SHTLN
166: 2050   .  .  FORMAT(1X,'SHOT LINE NOS.: ',A6,9(' ',A6))
167:        .  END IF
168:        .  WRITE(10,2003) PLACE
169: 2003   .  FORMAT(1X,'INSTITUTION RECD TAPE FROM',T40,5A6)
170:        .  WRITE(10,2004) IDOC
171: 2004   .  FORMAT(1X,'DOCUMENTATION CODE; 1=YES',T40,A3)
172:        .  WRITE(10,2005) RDATE
173: 2005   .  FORMAT(1X,'DATE ARCHIVE TAPE RECEIVED',T40,3A3)
174:        .  WRITE(10,2006) IDATE
175: 2006   .  FORMAT(1X,'DATE ARCHIVED',T40,3A3)
176:        .  WRITE(10,2007) JDATE
177: 2007   .  FORMAT(1X,'DATE LAST UPDATED',T40,3A3)
178:        .  WRITE(10,2008) KDATE
179: 2008   .  FORMAT(1X,'DATE LAST ACCESSED',T40,3A3)
180:        .  WRITE(10,2009)
181: 2009   .  FORMAT(1X,'************* MINIMUM & MAXIMUM VALUES OF ',
182:       +.  'KEYWORDS *************')
183:        .  WRITE(10,2010) INUMMN
184: 2010   .  FORMAT(1X,'INSTRUMENT NUMBER',T40,I6)
185:        .  WRITE(10,2011) ENUMMN,ENUMMX
186: 2011   .  FORMAT(1X,'EVENT NUMBERS',T40,2I6)
187:        .  CSEC=DSTMIN
188:        .  CALL CNTITM
189:        .  WRITE(10,2012) TRAY
190: 2012   .  FORMAT(1X,'MINIMUM DATA START TIME',T40,7I4)
191:        .  CSEC=DSTMAX
192:        .  CALL CNTITM
193:        .  WRITE(10,2013) TRAY
194: 2013   .  FORMAT(1X,'MAXIMUM DATA START TIME',T40,7I4)
195:        .  WRITE(10,2014) EXPLMN,EXPLMX
196: 2014   .  FORMAT(1X,'EXPLOSIVE TYPES',T40,2I2)
197:        .  WRITE(10,2015) DEPTMN
198: 2015   .  FORMAT(1X,'WATER DEPTH AT INSTRUMENT',T40,I6)
199:        .  WRITE(10,2016) DEPEMN,DEPEMX
200: 2016   .  FORMAT(1X,'WATER DEPTHS AT EVENT',T40,2I6)
201:        .  WRITE(10,2017) IDEPMN
202: 2017   .  FORMAT(1X,'INSTRUMENT DEPTH',T40,I6)
203:        .  WRITE(10,2018) EDEPMN,EDEPMX
204: 2018   .  FORMAT(1X,'EVENT DEPTHS',T40,2I6)
205:        .  WRITE(10,2019) ICHNMN,ICHNMX
206: 2019   .  FORMAT(1X,'#S OF CHANNELS',T40,2I3)
207:        .  WRITE(10,2020) TYPEMN,TYPEMX
208: 2020   .  FORMAT(1X,'EVENTS TYPES',T40,2I2)
209:        .  WRITE(10,2021) SIZEMN,SIZEMX
210: 2021   .  FORMAT(1X,'EVENT SIZES',T40,2F10.4)
211:        .  WRITE(10,2022) RANGMN,RANGMX
212: 2022   .  FORMAT(1X,'RANGES',T40,2F10.4)
213:        .  WRITE(10,2023) ILATMN
214: 2023   .  FORMAT(1X,'INSTRUMENT LATITUDE',T40,F10.4)
215:        .  WRITE(10,2024) ILONMN
216: 2024   .  FORMAT(1X,'INSTRUMENT LONGITUDE',T40,F10.4)
217:        .  WRITE(10,2025) ELATMN,ELATMX
218: 2025   .  FORMAT(1X,'EVENT LATITUDE RANGE',T40,2F10.4)
219:        .  WRITE(10,2026) ELONMN,ELONMX
220: 2026   .  FORMAT(1X,'EVENT LONGITUDE RANGE',T40,2F10.4)
221:        .  WRITE(10,2027)
222: 2027   .  FORMAT(1X,'***** TAPE HEADER FILE CONTENTS *****')
223:        .  WRITE(10,2028) IID,DNAME
224: 2028   .  FORMAT(1X,'INSTR. #',T20,I4,'    DESIGNER',T40,9A6)
225:        .  WRITE(10,2029) EXCODE,NFILES
226: 2029   .  FORMAT(1X,'EXPERIMENT:',T20,3A6,A2,'   # OF EVENTS:',
227:       +.  T60,I6)
228:        .  CSEC=TDST
229:        .  CALL CNTITM
230:        .  WRITE(10,2030) TRAY
231: 2030   .  FORMAT(1X,'TAPE DATA START TIME',T40,7I4)
232:        .  CSEC=TDET
233:        .  CALL CNTITM
234:        .  WRITE(10,2031) TRAY
235: 2031   .  FORMAT(1X,'TAPE DATA END TIME',T40,7I4)
236:        END LOOP
237:        STOP POO
238:        END
```

```
C
      NAME DVIEW
      INTEGER IBUF(4096),JBUF(4096)
      IKC=1
      IREC=0
      NF=1
C PROCESS TAPE HEADER FILE
      LOOP(2)
      . BUFFER IN(4,IBUF,B,256,IS,IL)
      . CALL STATUS(4)
      . IREC=IREC+1
      . WRITE(10,1001) IREC,IL
1001  . FORMAT(' RECORD NO.',I6,'  LENGTH',I6)
      . EXIT LOOP IF(IS.EQ.3)
      . WRITE(10,1000) (IBUF(J),J=1,54)
1000  . FORMAT(27A3)
      END LOOP
      WRITE(3,100)
100   FORMAT(' ENTER NO. OF FILES TO LOOK AT')
      READ(0,) NFILES
      WRITE(3,200)
200   FORMAT(' ENTER COMPONENT NUMBER TO DISPLAY:')
      READ(0,) IK
      LOOP(NFILES)
      . DO
      . . BUFFER IN(4,IBUF,B,4096,IS,IL)
      . . CALL STATUS(4)
      . . IREC=IREC+1
      . . WRITE(10,1001) IREC,IL
      . . IF(IL.EQ.0)
      . . . NF=NF+1
      . . END IF
      . UNTIL(IL.EQ.256)
      . NREC=IBUF(71)
      . NCOMP=IBUF(39)
      . IF(IBUF(1).LE.99) NCOMP=2
      . LOOP
      . . EXIT LOOP IF(IKC.EQ.IK)
      . . LOOP(NREC)
      . . . BUFFER IN(4,JBUF,B,4096,IS,IL)
      . . END LOOP
      . . IKC=IKC+1
      . END LOOP
      . WRITE(10,300) NF,IBUF(3),IK
300   . FORMAT(' FILE NO.',I6,'  EVENT NO.',I6,'  COMPONENT NO.',I6)
      . LOOP(NREC)
      . . BUFFER IN(4,IBUF,B,4096,IS,IL)
      . . CALL STATUS(4)
      . . IREC=IREC+1
      . . WRITE(10,1001) IREC,IL
      . . CALL CNVNEG(IBUF,4096)
      . . FOR L=1,51
      . . . LL=80*(L-1) + 1
      . . . WRITE(10,2001) (IBUF(I),I=LL,LL+79)
2001  . . . FORMAT(10(/8I10))
      . . END FOR
      . . WRITE(10,"(/2I8)") (IBUF(I),I=4081,4096)
      . END LOOP
      . IKC=1
      END LOOP
      STOP
      END
```

```
 1:       NAME RECLEN
 2:       INTEGER JBUF(11000)
 3:       I=0
 4:       NF=1
 5:       WRITE(3,"(' ENTER # FILES')")
 6:       READ (0,) NFILES
 7:       LOOP
 8:       .  BUFFER IN(4,JBUF,B,11000,IS,IL)
 9:       .  CALL STATUS(4)
10:       .  I=I+1
11:       .  EXIT LOOP IF(IS.GE.4)
12:       .  WRITE(3,"(' FILE NO. ',I4,'  RECNO,LENGTH',2I6)") NF,I,IL
13:       .  IF(IL.EQ.0)
14:       .  .  NF=NF+1
15:       .  .  WRITE(3,"(' ')")
16:       .  END IF
17:       .  IF(IL.EQ.256) WRITE(10,"(' EVENT NO.',I6)") JBUF(3)
18:       .  IF(NF.GT.NFILES) GOTO 999
19:       END LOOP
20:       WRITE(3,"(' NO. OF EOFS',I5)") NF
21:       STOP
22: 999   STOP NF
23:       END
```

```
C  PROGRAM B I A R C H TO READ AND REWRITE ROSE FORMATTED SEISMIC DATA
C  FROM TAPE AND TO ADD TO THE ARCHIVE CATALOG AREA.
C
C  WRITTEN BY SHARON LATRAILLE   HIG 363 X7796
C     LAST UPDATED 3/30/83 (REV 7)
C   TO RUN USE MACRO M<BIARC ;TO COMPILE USE J.BIARCH
C
C
      NAME BIARCH
      INTEGER IC(24),RERR,ELOER,BUB,SAMP,FNUM,INUM,ENUM
      INTEGER WDEPI,WDEPE,IDEP,EDEP,ICHN,CCODE,HDLFN
      INTEGER CATBUF(224),FHDR(260),IBUF(4096),DBUF(54)
      INTEGER RDATE(3),IDATE(3),IDAT1(3),EXPL,TYPE,MBUF(224)
      INTEGER CATLFN,TAPE,OUTF,DEPIMX,DEPIMN,DEPEMX,DEPEMN
      INTEGER ENUMMX,ENUMMN,EXPLMX,EXPLMN,ICHNMX
      INTEGER ICHNMN,TYPEMX,TYPEMN,EDEPMX,EDEPMN,REP
      INTEGER THBUF(224),TRAY1(7),TRAY(7),ITIME(3)
      INTEGER SBEG(10),SEND(10)
      REAL ILAT,ILON,ILATMX,ILATMN,ILONMX,ILONMN
      INTEGER*1 ICRAY(162),DCBUF(162)
      INTEGER*6 SBT,CSEC,DST,DSTMAX,DSTMIN,TDST,TDET,ISTATS(2)
      DIMENSION PLACE(5),DNAME(9),EXCODE(4)
      DIMENSION C(8),COMBUF(24),SHTLN(10),A(4)
      EQUIVALENCE (IBUF(1),ICRAY(1)),(DBUF,DCBUF)
C
      EQUIVALENCE (CATBUF(1),JTYPE),(CATBUF(2),ITAP),(CATBUF(3),INUM)
      EQUIVALENCE (CATBUF(4),ENUM),(CATBUF(5),DST),(CATBUF(7),SBT)
      EQUIVALENCE (CATBUF(9),SIZE),(CATBUF(11),RANGE)
      EQUIVALENCE (CATBUF(13),ILAT),(CATBUF(15),ILON),(CATBUF(17),ELAT)
      EQUIVALENCE (CATBUF(19),ELON),(CATBUF(21),EXPL),(CATBUF(22),WDEPI)
      EQUIVALENCE (CATBUF(23),WDEPE),(CATBUF(24),IDEP),(CATBUF(25),EDEP)
      EQUIVALENCE (CATBUF(26),ICHN),(CATBUF(27),TYPE)
      EQUIVALENCE (CATBUF(28),RERR),(CATBUF(29),ELOER),(CATBUF(30),BUB)
      EQUIVALENCE (CATBUF(31),SAMP),(CATBUF(32),NWDS)
      EQUIVALENCE (CATBUF(33),FNUM),(CATBUF(34),NREC),(CATBUF(35),NSAM)
      EQUIVALENCE (CATBUF(36),IDEL),(CATBUF(37),IDATE)
      EQUIVALENCE (CATBUF(40),IDAT1),(CATBUF(43),IC)
      EQUIVALENCE (CATBUF(67),SHTLIN),(CATBUF(69),CCODE)
      COMMON/ITCM/TRAY,CSEC,JULD
      COMMON/AVG/ IBUF,MEAN
      DATA TAPE/4/,OUTF/20/,CATLFN/30/,HDLFN/40/,C(1)/'V'/,C(2)/'R'/,
     +C(3)/'T'/,C(4)/'P'/,C(5)/'H1'/,C(6)/'H2'/,C(7)/'W'/,C(8)/'TI'/,
     +REP/10/
      CALL BTIME
      CALL DATE(IDATE)
      CALL TIME(ITIME)
      FOR I=1,3
      .  IDAT1(I)=IDATE(I)
      END FOR
      OPEN OUTF
      OPEN CATLFN
      OPEN TAPE
C FIND POSITION OF START OF TAPE HEADERS IN CATALOG
      CALL DPOS(CATLFN,2)
      CALL BUFIN(CATLFN,MBUF,112,IEOF)
      ITHEAD=MBUF(112)
C ZERO IC ARRAY AND SHTLN ARRAY
      FOR I=1,24
      .  IC(I)=0
      END FOR
      FOR I=1,10
      .  SHTLN(I)="      "
      END FOR
C
      WRITE(3,7950)
7950  FORMAT(1X,'Program ARCHIVE, REV. 6, Last modified 5/4/82',
     +/' ** CHECK SOURCE IF HAVE MIXTURE OF SHOTS & QUAKES')
C
C Read HIG header information
      READ(HDLFN,8000) ITAP,IDSH,IID1,NSLN
8000  FORMAT(T50,I3,/T50,I3,/T50,I5,/T50,I3)
      FOR J=1,NSLN
      .  READ(HDLFN,8001) SHTLN(J),SBEG(J),SEND(J)
8001  .  FORMAT(T50,A6,2I5)
      END FOR
      IEVST=SBEG(1)
      READ(HDLFN,8002) PLACE,IDOC,RDATE
8002  FORMAT(T50,5A6,/T50,I3,/T50,3A3)
      READ(HDLFN,8003) KRA,IDSH1,IKK
8003  FORMAT(T50,I3,2(/T50,I3))
```

```
 81:       IF(IKK.NE.0)
 82:       .  READ(HDLFN,8004) KDC,IOVRD,JTC
 83: 8004  .  FORMAT(T50,I3,/T50,I4,/T50,I3)
 84:       .  IF(JTC.EQ.1)
 85:       .  .  READ(HDLFN,8022) TDST,TDET
 86: 8022  .  .  FORMAT(2I14)
 87:       .  END IF
 88:       END IF
 89: C ENCODE HIG HEADER INFO INTO TAPE HEAD BUFFER
 90:       JTYPE=2
 91:       ENCODE(84,7500,THBUF) JTYPE,ITAP,NSLN,(PLACE(I),I=1,5),IDOC,
 92:      +(RDATE(I),I=1,3),(IDATE(I),I=1,3),(IDATE(I),I=1,3),
 93:      +(IDATE(I),I=1,3)
 94: 7500  FORMAT(I3,2I6,5A6,13A3)
 95:       ENCODE(3,7501,THBUF(160)) IDSH
 96: 7501  FORMAT(I3)
 97: C ********** BEGIN TAPE PROCESSING ********************
 98: C
 99:       BUFFER IN(TAPE,IBUF,B,256,ISTAT,ILEN)
100:       CALL STATUS (TAPE)
101:       IF (ISTAT.EQ.3)
102:       .  WRITE(3,7601)
103: 7601  .  FORMAT(1X,'EOF ON INITIAL TAPE HEADER READ')
104:       END IF
105: C WRITE TAPE HEADER FILE TO ARCHIVE FILE
106: C
107:       BUFFER OUT (OUTF,IBUF,B,256,ISTAT,ILEN)
108:       ENDFILE OUTF
109:       J=1
110:       FOR I=1,162
111:       .  K=MOD(I,3)
112:       .  IF(K.NE.1)
113:       .  .  DCBUF(J)=ICRAY(I)
114:       .  .  J=J+1
115:       .  END IF
116:       END FOR
117: C DECODE TAPE HEADER
118: C
119: C DECODE NUMBER OF FILES
120:       KCHAN=0
121: 40    DECODE(9,"(BN,I9)",DBUF(34)) NFILES
122: C DECODE IID
123:       DECODE(4,"(BN,I4)",DBUF) IID
124:       DECODE(55,8005,DBUF(2)) DNAME
125: 8005  FORMAT(1X,9A6)
126:       DECODE(20,8006,DBUF(21)) EXCODE
127: 8006  FORMAT(3A6,A2)
128:       DECODE(12,8010,DBUF(27))ISYR,ISMO,ISDA,ISHR,ISMIN
129:  8010 FORMAT(2X,5I2)
130:       DECODE(10,8020,DBUF(31))IFYR,IFMO,IFDA,IFHR,IFMIN
131:  8020 FORMAT (5I2)
132:       WRITE(3,8040) IID,NFILES
133:  8040 FORMAT(1X,"INSTRUMENT ID  ",I4,/," # FILES  ",I4)
134:       IF(IOVRD.NE.0)
135:       .  NFILES=IOVRD
136:       .  WRITE(3,8041) NFILES
137: 8041  .  FORMAT(' NO. FILES SET TO: ',I6)
138:       END IF
139:       WRITE(3,8045) DNAME
140: 8045  FORMAT(1X,'DESIGNER NAME & ADDRESS: ',9A6)
141:       WRITE(3,8046) EXCODE
142: 8046  FORMAT(1X,3A6,A2,' EXPERIMENT')
143:       WRITE(3,8050)ISYR,ISMO,ISDA,ISHR,ISMIN
144: 8050  FORMAT(1X,"START TIME  ",5(1X,I2))
145:       WRITE(3,8060)IFYR,IFMO,IFDA,IFHR,IFMIN
146: 8060  FORMAT(1X,"END TIME    ",5(1X,I2))
147:       TRAY(1)=ISYR
148:       TRAY(2)=ISMO
149:       TRAY(3)=ISDA
150:       TRAY(4)=ISHR
151:       TRAY(5)=ISMIN
152:       TRAY(6)=0
153:       TRAY(7)=0
154:       CALL ITMCNT
155:       TDST=CSEC
156:       TRAY(1)=IFYR
157:       TRAY(2)=IFMO
158:       TRAY(3)=IFDA
159:       TRAY(4)=IFHR
160:       TRAY(5)=IFMIN
```

```
161:       CALL ITMCNT
162:       TDET=CSEC
163: C Encode tape header info into current HIG tape header buffer
164: C
165:       ENCODE(174,7600,THBUF(95)) IID,DNAME,EXCODE,NFILES,TDST,TDET,SHTLN
166: 7600  FORMAT(I4,12A6,A2,I6,I14,1X,I14,1X,10A6)
167: C
168: C SAVE OFF TAPE HEADER RECORDS
169: C
170:       INEXT=ITHEAD
171:       CALL DPOS(CATLFN,INEXT)
172:       LOOP
173:       .  ISAV=0
174:       .  CALL BUFIN(CATLFN,MBUF,224,IEOF)
175:       .  EXIT LOOP IF(IEOF.GE.3)
176: C IF REARCHIVING, DONT SAVE TAPE HEADER RECORD FOR THIS TAPE
177:       .  IF(KRA.EQ.1)
178:       .  .  DECODE(9,7550,MBUF) KTAP
179: 7550  .  .  FORMAT(3X,I6)
180:       .  .  DECODE(6,7553,THBUF(95)) INM
181: 7553  .  .  FORMAT(I4,2X)
182:       .  .  DECODE(3,7501,MBUF(160)) IDASH
183:       .  .  DECODE(6,7551,MBUF(121)) NF
184: 7551  .  .  FORMAT(I6)
185:       .  .  IF(KTAP.EQ.ITAP)
186:       .  .  .  IF(IDSH1.EQ.IDASH.AND.NFILES.EQ.NF.AND.IID.EQ.INM)
187:       .  .  .  .  WRITE(3,7552) ITAP,IDSH1,IID,NFILES,KTAP,IDASH,INM,NF
116,88: 7552  .  .  .  .  FORMAT(1X,' Input tape,instrument,#files:',I6,'-',I2,2
189:      +.  .  .  .  /'   catalog file tape,instrument,#files:',I6,'-',I2,2I6)
190:       .  .  .  .  WRITE(3,7555)
191: 7555  .  .  .  .  FORMAT(' We have a match;Execute replacement ARCHIVE',
192:      +.  .  .  .  ' after saving remaining headers')
193:       .  .  .  .  INEXT=MBUF(153)
194:       .  .  .  .  ISAV=1
195:       .  .  .  END IF
196:       .  .  END IF
197:       .  END IF
198:       .  IF(ISAV.EQ.0)
199:       .  .  BUFFER OUT(11,MBUF,B,224,MSTAT,MLEN)
200:       .  .  CALL STATUS(11)
201:       .  END IF
202:       END LOOP
203:       REWIND 11
204:       THBUF(153)=INEXT
205:       CALL DPOS(CATLFN,INEXT)
206:       WRITE(3,1050)
207: 1050  FORMAT(/1X,"Begin reading and writing event header records")
208: C
209: C*****************************************************************
210: C READ/WRITE FILE HEADER RECORDS AND DATA
211: C
212:       JTYPE=1
213:       I=0
214:       IQ=0
215:       LOOP(NFILES)
216:       .  I=I+1
217:       .  DO
218:       .  .  BUFFER IN (TAPE,FHDR,B,260,IFST,ILEN)
219:       .  .  CALL STATUS (TAPE)
220:       .  .  IF(IFST.EQ.3)
221:       .  .  .  WRITE(3,7602)
222: 7602  .  .  .  FORMAT(1X,'EOF ENCOUNTERED')
223:       .  .  END IF
224:       .  .  EXIT LOOP IF(IFST.GE.4)
225:       .  UNTIL(ILEN.EQ.256)
226:       .  CALL CNVNEG(FHDR,256)
227:       .  BUFFER OUT(OUTF,FHDR,B,256,IST,ILEN)
228:       .  CALL STATUS (OUTF)
229:       .  FOR J=1,7
230:       .  .  TRAY(J)=FHDR(J+3)
231:       .  END FOR
232:       .  CALL ITMCNT
233:       .  INUM =FHDR(1)
234:       .  ENUM =FHDR(3)
235:       .  DST =CSEC
236:       .  SIZE =(10.**(FHDR(36)/1000.))/1000.
237:       .  IF(FHDR(2).EQ.1)
238:       .  .  SIZE =FLOAT(FHDR(36))/10.
239:       .  END IF
240:       .  ISIZE=IFIX(SIZE)
```

```
241:       .  RANGE =FLOAT(FHDR(11))*1000.+FLOAT(FHDR(12))
242:      +.  +FLOAT(FHDR(13))/1000.
243:       .  A(1)=FLOAT(FHDR(16))/1000.
244:       .  A(2)=FLOAT(FHDR(18))/1000.
245:       .  A(3)=FLOAT(FHDR(22))/1000.
246:       .  A(4)=FLOAT(FHDR(24))/1000.
247:       .  ILAT =ISIGN(1,FHDR(15))*(ABS(FLOAT(FHDR(15)))+ABS(A(1)))
248:       .  ILON =ISIGN(1,FHDR(17))*(ABS(FLOAT(FHDR(17)))+ABS(A(2)))
249:       .  ELAT =ISIGN(1,FHDR(21))*(ABS(FLOAT(FHDR(21)))+ABS(A(3)))
250:       .  ELON =ISIGN(1,FHDR(23))*(ABS(FLOAT(FHDR(23)))+ABS(A(4)))
251:       .    IF(RANGE.LT.0.0) CALL RANGER(ILAT,ILON,ELAT,ELON,RANGE)
252:       .  EXPL =FHDR(35)
253:       .  WDEPI =FHDR(20)
254:       .  WDEPE =FHDR(27)
255:       .  IDEP =FHDR(19)
256:       .  EDEP =FHDR(26)
257:       .  ICHN =FHDR(39)
258:       .  IF(KCHAN.NE.0) ICHN=KCHAN
259:       .  CCODE=FHDR(60)
260:       .  IF(ICHN.GT.10)
261:       .  .  CCODE=1
262:       .  END IF
263:       .  TYPE =FHDR(2)
264:       .  IF(FHDR(28).GT.0)
265:       .  .  FOR J=1,7
266:       .  .  .  TRAY(J)=FHDR(J+27)
267:       .  .  END FOR
268:       .  .  CALL ITMCNT
269:       .  .  SBT=CSEC
270:       .  ELSE
271:       .  .  SBT=DST
272:       .  END IF
273:       .  RERR=FHDR(14)
274:       .  ELOER=FHDR(25)
275:       .  BUB=FHDR(37)
276:       .  SAMP=FHDR(38)
277:       .  NWDS=FHDR(40)
278:       .  FNUM=FHDR(41)
279:       .  NREC=FHDR(71)
280:       .  NSAM=FHDR(72)
281:       .  IDEL=0
282:       .  FOR JK=1,NSLN
283:       .  .  IF(ENUM.GE.SBEG(JK).AND.ENUM.LE.SEND(JK))
284:       .  .  .  SHTLIN=SHTLN(JK)
285:       .  .  END IF
286:       .  END FOR
287: C FOR LESS THAN 11 CHANNELS
288:       .  IF(ICHN.LT.11)
289:       .  .  FOR K=1,ICHN
290:       .  .  .  IC(K)=FHDR(41+K*20)
291:       .  .  END FOR
292:       .  END IF
293: C FOR MORE THAN 11 CHANNELS
294:       .  IF(ICHN.GE.11)
295:       .  .  IF(CCODE.EQ.0)
296:       .  .  .  HEAD=(FLOAT(ICHN)-10.)/12.
297:       .  .  .  NHDRS=IFIX(HEAD)
298:       .  .  .  IF(HEAD-FLOAT(NHDRS).GT.0.0)
299:       .  .  .  .  NHDRS=NHDRS+1
300:       .  .  .  END IF
301:       .  .  .  FOR K=1,10
302:       .  .  .  .  IC(K)=FHDR(41+K*20)
303:       .  .  .  END FOR
304:       .  .  .  FOR IH=1,NHDRS
305:       .  .  .  .  DO
306:       .  .  .  .  .  BUFFER IN(TAPE,FHDR,B,260,IFST,ILEN)
307:       .  .  .  .  .  CALL STATUS(TAPE)
308:       .  .  .  .  UNTIL(ILEN.EQ.256)
309:       .  .  .  .  CALL CNVNEG(FHDR,256)
310:       .  .  .  .  BUFFER OUT(OUTF,FHDR,B,256,IST,ILEN)
311:       .  .  .  .  L=10*IH+IH
312:       .  .  .  .  LL=L+12
313:       .  .  .  .  FOR K=L,LL
314:       .  .  .  .  .  IC(L)=FHDR(1+(IH-1)*20)
315:       .  .  .  .  END FOR
316:       .  .  .  END FOR
317:       .  .  END IF
318:       .  .  IF(CCODE.EQ.1)
319:       .  .  .  IC(1)=FHDR(61)
320:       .  .  END IF
```

```
321:       . END IF
322: C WRITE VARIABLES, INCLUDING KEYWORDS, TO CATBUF
323: C
78           324:       . CALL BUFOUT(CATLFN,CATBUF,112,IEOF)
325:       . CALL DSTAT(CATLFN,ISTATS,ICRA)
326: C INITIALIZE MIN,MAX VALUES OF KEYWORDS
327: C     CHECK FOR QUAKES
328: C
329:       . IF(TYPE.EQ.1)
330:       . . I=I-1
331:       . . IQ=IQ+1
332:       . . GO TO 777
333:       . END IF
334:       . IF(I.EQ.1)
335:       . . INUMMX=INUM
336:       . . INUMMN=INUM
337:       . . ENUMMX=ENUM
338:       . . ENUMMN=ENUM
339:       . . DSTMAX=DST
340:       . . DSTMIN=DST
341:       . . EXPLMX=EXPL
342:       . . EXPLMN=EXPL
343:       . . DEPIMX=WDEPI
344:       . . DEPIMN=WDEPI
345:       . . DEPEMX=WDEPE
346:       . . DEPEMN=WDEPE
347:       . . IDEPMX=IDEP
348:       . . IDEPMN=IDEP
349:       . . EDEPMX=EDEP
350:       . . EDEPMN=EDEP
351:       . . ICHNMX=ICHN
352:       . . ICHNMN=ICHN
353:       . . TYPEMX=TYPE
354:       . . TYPEMN=TYPE
355:       . . SIZEMX=SIZE
356:       . . SIZEMN=SIZE
357:       . . RANGMX=RANGE
358:       . . RANGMN=RANGE
359:       . . ILATMX=ILAT
360:       . . ILATMN=ILAT
361:       . . ILONMX=ILON
362:       . . ILONMN=ILON
363:       . . ELATMX=ELAT
364:       . . ELATMN=ELAT
365:       . . ELONMX=ELON
366:       . . ELONMN=ELON
367:       . END IF
368: C NOW COMPARE NEW VALUES OF KEYWORDS WITH MIN,MAX AND GET NEW MIN,MAX
369: C
370:       . INUMMX=MAX0(INUMMX,INUM)
371:       . INUMMN=MIN0(INUMMN,INUM)
372:       . ENUMMX=MAX0(ENUMMX,ENUM)
373:       . ENUMMN=MIN0(ENUMMN,ENUM)
374:       . DSTMAX=MAX2(DSTMAX,DST)
375:       . DSTMIN=MIN2(DSTMIN,DST)
376:       . EXPLMX=MAX0(EXPLMX,EXPL)
377:       . EXPLMN=MIN0(EXPLMN,EXPL)
378:       . DEPIMX=MAX0(DEPIMX,WDEPI)
379:       . DEPIMN=MIN0(DEPIMN,WDEPI)
380:       . DEPEMX=MAX0(DEPEMX,WDEPE)
381:       . DEPEMN=MIN0(DEPEMN,WDEPE)
382:       . IDEPMX=MAX0(IDEPMX,IDEP)
383:       . IDEPMN=MIN0(IDEPMN,IDEP)
384:       . EDEPMX=MAX0(EDEPMX,EDEP)
385:       . EDEPMN=MIN0(EDEPMN,EDEP)
386:       . ICHNMX=MAX0(ICHNMX,ICHN)
387:       . ICHNMN=MIN0(ICHNMN,ICHN)
388:       . TYPEMX=MAX0(TYPEMX,TYPE)
389:       . TYPEMN=MIN0(TYPEMN,TYPE)
390:       . SIZEMX=AMAX1(SIZEMX,SIZE)
391:       . SIZEMN=AMIN1(SIZEMN,SIZE)
392:       . RANGMX=AMAX1(RANGMX,RANGE)
393:       . RANGMN=AMIN1(RANGMN,RANGE)
394:       . ILATMX=AMAX1(ILATMX,ILAT)
395:       . ILATMN=AMIN1(ILATMN,ILAT)
396:       . ILONMX=AMAX1(ILONMX,ILON)
397:       . ILONMN=AMIN1(ILONMN,ILON)
398:       . ELATMX=AMAX1(ELATMX,ELAT)
399:       . ELATMN=AMIN1(ELATMN,ELAT)
400:       . ELONMX=AMAX1(ELONMX,ELON)
```

```
401:       .  ELONMN=AMIN1(ELONMN,ELON)
402: C COPY DATA RECORDS TO DISC FILE
403: 777   .  N=ICHN*NREC
404:       .  NTAL=N
405:       .  WRITE(3,4112) ENUM,FNUM
406: 4112  .  FORMAT(1X,'PROCESSING EVENT NO.  ',I6,' FILE NO. ',I6)
407:       .  NIN=0
408:       .  ILOOP=0
409:       .  LOOP(N)
410:       .  .  BUFFER IN (TAPE,IBUF,B,4096,IST,ILEN)
411:       .  .  CALL STATUS (TAPE)
412:       .  .  CALL CNVNEG(IBUF,4096)
413:       .  .  NTAL=NTAL-1
414: C NOW TAKE OUT DC OFFSET
415:       .  .  IF(KDC.EQ.1)
416:       .  .  .  ILOOP=ILOOP+1
417:       .  .  .  IRNO=1+NIN*NREC
418:       .  .  .  IF(ILOOP.EQ.IRNO)
419:       .  .  .  .  CALL AVER
420:       .  .  .  .  NIN=NIN+1
421:       .  .  .  END IF
422:       .  .  .  FOR J=1,4096
423:       .  .  .  .  IBUF(J)=IBUF(J)-MEAN
424:       .  .  .  END FOR
425:       .  .  END IF
426:       .  .  BUFFER OUT(OUTF,IBUF,B,4096,ISTAT,ILEN)
427:       .  END LOOP
428:       .  IF(NTAL.NE.0)
429:       .  .  WRITE(3,1123) ENUM,N,NTAL
430: 1123  .  .  FORMAT(' EVENT',I6,' WITH',I6,' RECORDS TO PROCESS WAS SHORT',
431:      +.  .  ',I6,' RECORDS')
432:       .  END IF
433:       .  ENDFILE OUTF
434:       END LOOP
435:       ENDFILE OUTF
436: C ADD TAPE HEADER RECORDS BACK ON TO EVENT CATALOG
437: C
438:       CALL DSTAT(CATLFN,ISTATS,ICRA)
439:         IF(KRA.EQ.1)
440:       .    CALL DPOS(CATLFN,ITHEAD)
441:         END IF
442:       LOOP
443:       .  BUFFER IN(11,MBUF,B,224,MSTAT,MLEN)
444:       .  CALL STATUS(11)
445:       .  EXIT LOOP IF(MSTAT.GE.3)
446:       .  CALL BUFOUT(CATLFN,MBUF,224,IEOF)
447:       END LOOP
448:       CALL DSTAT(CATLFN,ISTATS,ILAST)
449:         IF(KRA.NE.1)
450:       .  CALL DPOS(CATLFN,2)
451:       .  CALL BUFIN(CATLFN,MBUF,112,IEOF)
452:       .  MBUF(112)=ICRA
453:       .  CALL DPOS(CATLFN,2)
454:       .  CALL BUFOUT(CATLFN,MBUF,112,IEOF)
455:       .  WRITE(3,1121) ICRA
456: 1121  .  FORMAT(1X,'START RECORD FOR TAPE HEADERS NOW:',I6)
457:         END IF
458: C CHECK FOR INSTRUMENT KEYWORD VALUE MISMATCHES
459:       IF(ILATMX.NE.ILATMN)
460:       .  WRITE(3,2000)
461: 2000  .  FORMAT(1X,'INST. LATITUDES ENCODED NOT EQUAL FOR ALL EVENTS')
462:       END IF
463:       IF(INUMMN.NE.INUMMX)
464:       .  WRITE(3,2001)
465: 2001  .  FORMAT(1X,'INST # ENCODED NOT EQUAL FOR ALL EVENTS')
466:       END IF
467:       IF(DEPIMN.NE.DEPIMX)
468:       .  WRITE(3,2002)
469: 2002  .  FORMAT(1X,'WATER DEPTHS AT INST NOT SAME FOR ALL EVENTS')
470:       END IF
471:       IF(IDEPMN.NE.IDEPMX)
472:       .  WRITE(3,2003)
473: 2003  .  FORMAT(1X,'INST. DEPTHS NOT SAME FOR ALL EVENTS')
474:       END IF
475: C CHECK FOR NEGATIVE LATS AND LONS; REVERSE THEM
476: C
477:       IF(ELATMX.LT.0.0.AND.ELATMN.LT.0.0)
478:       .  TEMP=ELATMX
479:       .  ELATMX=ELATMN
480:       .  ELATMN=TEMP
```

```
      END IF
      IF(ELONMX.LT.0.0.AND.ELONMN.LT.0.0)
      .  TEMP=ELONMX
      .  ELONMX=ELONMN
      .  ELONMN=TEMP
      END IF
C ENCODE min,max KEYWORD values into HIG tape header buffer
C
      ENCODE(96,6000,THBUF(29)) INUMMN,ENUMMX,ENUMMN,DSTMAX,
     +DSTMIN,EXPLMX,EXPLMN,DEPIMN,DEPEMX,DEPEMN,
     +IDEPMN,EDEPMX,EDEPMN,ICHNMX,ICHNMN,TYPEMX,TYPEMN
6000  FORMAT(3I6,2I13,2I2,6I6,2I3,2I2,2X)
      ENCODE(102,6001,THBUF(61)) SIZEMX,SIZEMN,RANGMX,RANGMN,
     +ILATMN,ILONMN,ELATMX,ELATMN,ELONMX,ELONMN
6001  FORMAT(10F10.4,2X)
      CALL DPOS(CATLFN,ILAST)
      CALL BUFOUT(CATLFN,THBUF,224,IEOF)
      BUFFER OUT(11,THBUF,B,224,MSTAT,MLEN)
      ENDFILE 11
      ENDFILE CATLFN
      CLOSE CATLFN
C ***************************************************************
C PRINTED REPORT
C ***************************************************************
      IPG=1
      WRITE(REP,1) IPG
1     FORMAT(////T53,'ROSE ARCHIVE REPORT',T114,'PAGE NO. ',I3)
      WRITE(REP,2)
2     FORMAT(//T45,'* * * SUMMARY OF DATA ARCHIVED * * *' ,
     .T112,'RUN DATE,TIME')
        IF(KRA.NE.1)
      .  WRITE(REP,3) IDATE,ITIME
3     .  FORMAT(T111,3A3,', ',3A3)
        ELSE
      .    WRITE(REP,4) IDATE,ITIME
4     .    FORMAT(T52,'<<REPLACEMENT ARCHIVE>>',T111,3A3,', ',3A3)
        END IF
C ENCODE COMPONENT TYPES INTO COMBUF
      ALPHA='  '
      FOR K=1,24
      .  ENCODE(3,6777,COMBUF(K)) ALPHA
      END FOR
      FOR K=1,ICHN
      .  FOR J=1,8
      .  .  IF(IC(K).EQ.J)
      .  .  .  ENCODE(3,6777,COMBUF(K)) C(J)
6777  .  .  .  FORMAT(A3)
      .  .  END IF
      .  END FOR
      END FOR
C TRANSLATE START TIMES FOR PRINTING
      CSEC=DSTMIN
      CALL CNTITM
      FOR I=1,5
      .  TRAY1(I)=TRAY(I)
      END FOR
      CSEC=DSTMAX
      CALL CNTITM
      WRITE(REP,400) ITAP,IDSH,RDATE,EXCODE
400   FORMAT(//T5,'RARC TAPE # ',I5,'-',I3,T25,'DATE RECEIVED: ',
     +3A3,T82,'EXPERIMENT: ',3A6,A2)
      WRITE(REP,401) ISYR,ISMO,ISDA,ISHR,ISMIN
401   FORMAT(T5,'TAPE DATA START TIME: ',5I4)
      WRITE(REP,402) IFYR,IFMO,IFDA,IFHR,IFMIN
402   FORMAT(T5,'TAPE DATA END TIME  : ',5I4)
      WRITE(REP,403) IID,DNAME
403   FORMAT(/T5,'INSTRUMENT #: ',I4,T48,
     +'DESIGNER: ',9A6)
      WRITE(REP,404) IDOC,PLACE
404   FORMAT(T5,'DOCUMENTATION CODE (YES=1): ',A3,T48,
     +'INSTITUTION RECEIVED FROM: ',5A6)
      WRITE(REP,405) (COMBUF(I),I=1,10),ILAT,ILON
405   FORMAT(T5,'COMPONENTS 1-10 ONLY: ',10A2,T48,'INSTRUMENT LATITUDE:'
     +F8.4,T80,'INSTRUMENT LONGITUDE: ',F10.4)
      WRITE(REP,406) IDEP,WDEPI
406   FORMAT(T5,'INSTRUMENT DEPTH: ',I6,' M.',T48,
     +'WATER DEPTH AT INSTRUMENT: ',I4,' MSEC.')
      WRITE(REP,407) NFILES
407   FORMAT(T5,'NUMBER OF EVENTS: ',I4)
      WRITE(REP,408)
```

```
561: 408   FORMAT(//T5,'EVENT KEYWORD MINIMUM & MAXIMUM VALUES:')
562:       IF(NSLN.EQ.1)
563:       . WRITE(REP,304) TYPEMX,SHTLN(1)
564: 304   . FORMAT(/T20,'EVENT TYPE  ',I2,T48,'SHOT LINE #: ',A6)
565:       END IF
566:       IF(NSLN.GT.1)
567:       . WRITE(REP,325) TYPEMX,SHTLN
568: 325   . FORMAT(/T20,'EVENT TYPE ',I2,T48,'SHOT LINE #S: ',
569:      +. A6,9(' ',A6))
570:       END IF
571:       WRITE(REP,305) ENUMMN,ENUMMX
572: 305   FORMAT(T20,'EVENT #S  ',T52,I6,T60,' TO ',I6)
573:       WRITE(REP,306) (TRAY1(I),I=1,5),(TRAY(I),I=1,5)
574: 306   FORMAT(T20,'DATA START TIMES  ', 5I4,T60,' TO ',I6,4I4)
575:       WRITE(REP,308) ELATMN,ELATMX
576: 308   FORMAT(T20,'EVENT LATITUDES  ',T48,F10.3,T60,' TO ',F10.3)
577:       WRITE(REP,309) ELONMN,ELONMX
578: 309   FORMAT(T20,'EVENT LONGITUDES ',T48,F10.3,T60,' TO ',F10.3)
579:       WRITE(REP,310) EDEPMN,EDEPMX
580: 310   FORMAT(T20,'EVENT DEPTHS  ',T53,I5,T60,' TO  ',I5,' M')
581:       WRITE(REP,311) DEPEMN,DEPEMX
582: 311   FORMAT(T20,'WATER DEPTHS  ',T52,I6,T60,' TO ',I6,' M')
583:       WRITE(REP,312) SIZEMN,SIZEMX
584: 312   FORMAT(T20,'EVENT SIZES  ',T53,F5.1,T60,' TO  ',F5.1,' KG')
585:       IF(EXPLMN.EQ.EXPLMX) GO TO 10
586:       WRITE(REP,313) EXPLMN,EXPLMX
587: 313   FORMAT(T20,'EXPLOSIVE TYPES  ',T56,I2,T60,' TO ',I2)
588:       GO TO 15
589: 10    WRITE(REP,316) EXPLMN
590: 316   FORMAT(T20,'EXPLOSIVE TYPE  ',T48,I2)
591: 15    WRITE(REP,314) RANGMN,RANGMX
592: 314   FORMAT(T20,'RANGES-EVENT TO RCVR ',T52,F6.1,T59,' TO ',F6.1,' KM')
593:       WRITE(REP,318) IQ
594: 318   FORMAT(T20,'NUMBER OF EARTHQUAKES= ',I6)
595:       WRITE(REP,317)
596: 317   FORMAT(1H1)
597:       CLOSE OUTF
598:       CALL ETIME
599:       STOP YEA
600: 999     STOP EOF
601:       END

602:       SUBROUTINE AVER
603: C  THIS SUBROUTINE COMPUTES THE MEAN VALUE OF THE FIRST 200
604: C  DATA POINTS OF EACH CHANNEL IF DC OFFSET CORRECTION IS REQUESTED
605: C
606:       COMMON/AVG/ IBUF,MEAN
607:       INTEGER SUM,IBUF(4096)
608: C
609:       N=0
610:       SUM=0
611:       LOOP(20)
612:       . FOR I=1,10
613:       . . SUM=SUM+IBUF(I+N*10)
614:       . END FOR
615:       . IF(N.EQ.0)
616:       . . MEAN=SUM/10
617:       . END IF
618:       . IF(N.GT.0)
619:       . . MEAN=(MEAN+(SUM/10))/2
620:       . END IF
621:       . N=N+1
622:       END LOOP
623:       RETURN
624:       END
625:         SUBROUTINE RANGER(RLAT,RLON,SLAT,SLON,RAN)
626:       DR=3.141592654/180.
627:       DY=RLAT-SLAT
628:       DX=(RLON-SLON)*COS((RLAT+SLAT)*DR/2.)
629:       RAN=1.852*SQRT(DX*DX+DY*DY)*60.
630:         RETURN
631:       END
```

```
$MS
$PR MACRO  M < B I A R C - MACRO TO ARCHIVE ROSE DATA TO BINARY CATFILE
IF,(.NOT.(C.SPA.A&0)) $JU !GEN
$PR ENTER NAME OF HEADER FILE
$SR.IT #FIL
$AS 11=#FIL
$PR ENTER NAME OF DATA FILE
$SR.IT #FIL
$AS 20=#FIL
PR TAPE DRIVE 12
/PS 12
PR TAPE DRIVE 10
/PS 10
PR TAPE DRIVE 9
/PS 9
PR TAPE DRIVE 9
/PS 9
PR ENTER DRIVE NO. TO RESOURCE
$SR.IN #NUM
$JU !COPY
!GEN $PR ENTER NAME FOR DATA OUTPUT FILE
$SR.IT #FIL
$GE #FIL G200 M100000 P1
JE 318 !BIG
AS 20=#FIL
PR ENTER NAME OF HEADER FILE
SR.IT #HED
$ PR ENTER NAME FOR REPORT FILE
$ SR.IT #REP
$ GE #REP G=10
AS 10=REPORT
AS 11=THEAD
AS 6=T2
AS 40=#HED
AS 30=REVCAT.B
PR TAPE DRIVE 12
/PS 12
PR TAPE DRIVE 10
/PS 10
PR TAPE DRIVE 9
/PS 9
PR ENTER DRIVE NO. TO RESOURCE
$SR.IN #NUM
PR ENTER DENSITY 800 OR 1600
SR.IN #DEN
$PR LOAD TAPE TO BE ARCHIVED ON DRIVE-CHECK SCREEN FOR ROSTAP
IF (#DEN=800) JU !EIGHT
RS 4=ROSTAP 1600B 2C WA :#NUM
JU !EXCUT
!EIGHT RS 4=ROSTAP  800B 2C WA :#NUM
!EXCUT $1512ROSE*XBIWH
JE.P 37 !MAG
FR 4
$PR DISMOUNT TAPE
WI 10
CO T2 10
FR 10 40
RW 11
CO REPORT :6
CO REPORT :6
!COPY $PR WRITING TO BLANK TAPE? ENTER YES OR NO:
$SR.IT #BLK
$PR MOUNT ARCHIVE TAPE WITH WRITE RING-CHECK SCREEN FOR ARCTAP
IF, (#NUM=11) SR.N #NUM=9
RS 4=ARCTAP 1600B WR  WA :#NUM
IF,(#BLK="YES") $JU !WRIT
$$XADV
!WRIT $CO 11 4 BB TB=224 REC 1
$PR TAPE HEADER FILE JUST GENERATED NOW WRITTEN TO ARCTAP
WE 4
CO #FIL 4 BB TB=4144 ALL
$PR DATA WRITTEN TO ARCTAP
PR NOW DO CHECK READ OF ARCHIVE TAPE
PR REWIND TAPE? (ENTER RW) OR BACK UP? (BU):
$SR.IT #ANS
$IF (#ANS="RW") $JU !REW
$PR ENTER NUM OF FILES TO BACK (NUM OF EVENTS + 2):
$SR.IN #IRC
$BF 4 #IRC
```

```
81: $AR 4
82: $JU !ASSN
83: !REW $RW 4
84: !ASSN $AS 10=T2
85: XDISTAP
86: PR LIST T2 FOR CONTENTS OF TAPE HEADER
87: FR 4
88: $PR ARCHIVE PAU - DISMOUNT TAPE, LOG & STORE
89: PR ELIMINATE DATAOUT AREA
90: $ME
91: !BIG PR THAT'S TOO MANY CHARACTERS! TRY AGAIN.
92: JU !GEN
93: !MAG PR FATAL MAG TAPE ERROR: CHECK REPORT FILE FOR LINE # IN PROGRAM
94: $ME
```

```
      NAME HEDGEN
C ******* Program H E D G E N to generate data needed to build
C         the HIG ROSE archive header
C         The output of this program is an input file to XBHIROS
C ***************************************************************
      INTEGER*6 TDST,TDET,CSEC
      INTEGER RDATE(3),SBEG(10),SEND(10),TRAY(7)
      DIMENSION PLACE(5),SHTLN(10)
      COMMON/ITCM/TRAY,CSEC,JULD
C INPUT FROM THE TERMINAL
C
      WRITE(3,7950)
7950  FORMAT(1X,'Program HEDGEN, Last modified 4/27/82',
     +/' to generate the HIG ROSE header parameters & other info.')
      WRITE(3,8000)
8000  FORMAT(1X,'ENTER ARCHIVE TAPE # & DASH NUMBER :')
      READ(0,) ITAP,IDSH
      IDSH1=IDSH
      WRITE(3,7955)
7955  FORMAT(1X,'Enter instrument ID and number of shotlines: ')
      READ(0,) IID,NSLN
      WRITE(3,7967)
7967  FORMAT(' Please enter shotlines in the order in which they',
     +/' appear on the incoming tape or file'/)
      FOR J=1,NSLN
      . WRITE(3,7956)
7956  . FORMAT(1X,'ENTER SHOTLINE # (UP TO 6 CHAR; SLN1S): ')
      . READ(0,7960) SHTLN(J)
7960  . FORMAT(A6)
      . WRITE(3,7965) SHTLN(J)
7965  . FORMAT(1X,'ENTER BEG AND END SHOTS FOR SHOTLINE ',A6)
      . READ(0,) SBEG(J),SEND(J)
      END FOR
      WRITE(3,8002)
8002  FORMAT(1X,'Enter name of Institution tape received from:')
      READ(0,7001) PLACE
7001  FORMAT(5A6)
      WRITE(3,8003)
8003  FORMAT(1X,'ENTER DOCUMENTATION CODE; 0=NO,1=YES:')
      READ(0,)IDOC
      WRITE(3,8004)
8004  FORMAT(1X,'Enter date tape received as follows - ',
     +' DAY,MONTH,YEAR(LIKE 5 OCT 79:')
      READ(0,7002) RDATE
7002  FORMAT(3A3)
      WRITE(3,8011)
8011  FORMAT(1X,'IS THIS A RE-ARCHIVE? YES=1: ')
      READ(0,) KRA
      IF(KRA.EQ.1)
      . WRITE(3,7009)
7009  . FORMAT(' Enter old dash number as in Catalog:')
      . READ(0,) IDSH1
      END IF
      WRITE(3,7010)
7010  FORMAT(' If there is no DC offset to remove and no tape header',
     +/' parameters to override, enter 0:')
      READ(3,) IKK
      IF(IKK.GT.0)
      . WRITE(3,8007)
8007  . FORMAT(1X,'TAKE OUT DC OFFSET? YES=1:')
      . READ(0,) KDC
      . WRITE(3,1119)
1119  . FORMAT(1X,'OVERRIDE # OF FILES? IF NO, ENTER 0; ',
     +. /' IF SO, ENTER # OF FILES TO PROCESS:')
      . READ(0,) IOVRD
      . WRITE(3,1120)
1120  . FORMAT(' WANT TO ENTER NEW START & END TIMES? YES=1:')
      . READ(0,) JNE
      . TDST=0
      . TDET=0
      . IF(JNE.EQ.1)
      . . WRITE(3,8050)
8050  . . FORMAT(1X," Enter start time;as IYR,IMO,IDA,IHR,IMN:")
      . . READ(0,) (TRAY(I),I=1,5)
      . . TRAY(6)=0
      . . TRAY(7)=0
      . . CALL ITMCNT
      . . TDST=CSEC
      . . WRITE(3,8060)
```

```
      READ(0,) (TRAY(I),I=1,5)
      CALL ITMCNT
      TDET=CSEC
      END IF
      END IF
C WRITE ALL PARAMETERS TO HEADER FILE
      WRITE(40,9000) ITAP,IDSH,IID,NSLN
9000  FORMAT(' Tape number',T50,I3,/' Dash number',T50,I3,
     +/' Instrument ID',T50,I5,/' Number of shotlines',T50,I3)
      FOR J=1,NSLN
      WRITE(40,9001) J,SHTLN(J),SBEG(J),SEND(J)
9001  FORMAT(' Shotline',I3,' and shots',T50,A6,2I5)
      END FOR
      WRITE(40,9002) PLACE,IDOC,RDATE
9002  FORMAT(' Designer name and address',T50,5A6,
     +/' Documentation code',T50,I3,/' Date received',T50,3A3)
      WRITE(40,9003) KRA,IDSH1
9003  FORMAT(' Rearchiving? Yes=1',T50,I3,/' Old dash number-'
     +'if rearchiving',T50,I3)
      WRITE(40,9004) IKK
9004  FORMAT(' Deviations from standard archive? No=0',T50,I3)
      IF(IKK.NE.0)
      WRITE(40,9005) KDC,IOVRD
9005  FORMAT(' Take out DC offset? Yes=1',T50,I3,
     +/' Override number of files as in tape header?',
     +/'          No=0, Yes=new # of files',T50,I4)
      WRITE(40,9006) JNE,TDST,TDET
9006  FORMAT(' Override tape start and end times? Yes=1',T50,I3,
     +/' New start time (century msec)',T50,I14,
     +/' New end time   (   "      "  )',T50,I14)
      END IF
      STOP
      END
```

```
C PROGRAM TAPOS TO POSITION HIG ARCHIVE TAPES
C
C                    LAST MODIFIED 6/17/82   SLL
C
      NAME TAPOS
      INTEGER TAPE,JBUF(4096)
      DATA TAPE/4/
C
C
      ICNT=0
      LOOP
1     . BUFFER IN (TAPE,JBUF,B,4096,IS,IL)
      . CALL STATUS(TAPE)
      . ICNT=ICNT+1
      . IF (IS.GE.4) STOP EOT
      . IF(IL.EQ.0) GOTO 1
      . IF(IL.EQ.4096)
      . . IF(ICNT.EQ.1)
      . . . WRITE(3,1000)
1000  . . . FORMAT(' Data word;advancing to next header')
      . . END IF
      . . GO TO 1
      . END IF
      . IF(IL.EQ.256)
      . . WRITE(3,1002) JBUF(1),JBUF(3)
1002  . . FORMAT(' Instrument',I5,'   Event',I6)
      . . EXIT LOOP
      . END IF
      . IF(IL.EQ.224)
      . . CALL DECHDR(JBUF,NF,1)
      . . EXIT LOOP
      . END IF
      . IF(IL.NE.224.AND.IL.NE.256.AND.IL.NE.4096.AND.IL.NE.0)
      . . WRITE(3,1009) IL
1009  . . FORMAT(' WORD LENGTH ',I5,' NOT STANDARD ROSE FORMAT')
      . . STOP ERR
      . END IF
      END LOOP
      WRITE(3,1011)
1011  FORMAT(' Position to the end of the Nth archive; enter N:',
     +/'     or, rewind and start over; enter 0')
      READ(0,) NPOS
      IF(NPOS.EQ.0)
      . REWIND TAPE
      . GO TO 1
      END IF
      LCNT=0
      N=NPOS-1
      IF(NF.EQ.0) NF=999
      LOOP
      . IF(N.EQ.0) GO TO 2
      . DO
      . . BUFFER IN (TAPE,JBUF,B,4096,IS,IL)
      . . CALL STATUS(TAPE)
      . UNTIL(IL.EQ.224)
      . LCNT=LCNT+1
      . CALL DECHDR(JBUF,NF,LCNT)
      . WRITE(3,1008)
1008  . FORMAT(' Continuing to advance')
2     . LOOP(NF)
      . . DO
      . . . BUFFER IN (TAPE,JBUF,B,4096,IS,IL)
      . . . CALL STATUS(TAPE)
      . . . IF(IL.EQ.224)
      . . . . BACKSPACE TAPE
      . . . . EXIT LOOP
      . . . END IF
      . . UNTIL(IL.EQ.0)
      . END LOOP
      . EXIT LOOP IF(LCNT.GE.N)
      END LOOP
      WRITE(3,1010)
1010  FORMAT(' Tape positioned')
      STOP
      END
C
      SUBROUTINE DECHDR(THBUF,NFILES,IDASH)
      INTEGER THBUF(1),ENUMMN,ENUMMX
      DECODE(18,6001,THBUF(29)) INUM,ENUMMX,ENUMMN
6001  FORMAT(3I6)
```

```
      DECODE(84,6002,THBUF(95)) NFILES
 6002 FORMAT(78X,I6)
      DECODE(6,6003,THBUF(2)) ITAP
 6003 FORMAT(I6)
      WRITE(3,1004) ITAP,IDASH,INUM,ENUMMN,ENUMMX,NFILES
 1004 FORMAT(' TAPE',I4,' - ',I3,' : INSTRUMENT ',I4,
     +'  EVENTS',I6,'-',I6,'  # FILES:',I6)
      RETURN
      END
```

```
      NAME BHIROS
C
C  A PROGRAM TO CONVERT HIG DEMULTIPLEXED DATA AND CORFILE INFORMATION TO
C  ROSE ARCHIVE FORMAT AND WRITE BINARY CATALOG RECORDS.
C
C   WRITTEN BY SLL 10/10/79 WITH MUCH HELP FROM MIKE SIMPSON
C            LAST MODIFIED 5/4/82 SLL
C
C              ASSIGNMENTS: AS 10=REPORT FILE
C                           AS 11=TEMP HEADER STORAGE FILE (THEAD)
C                           AS 12=HDLFN(SEE HEDGEN)
C                           AS 15=ROSE TAPE HEADER FILE(RHEAD)
C                           AS 20=TEMPORARY ROSE ARCHIVE FILE HEADER RECO
C                           AS 30=CATALOG FILE (REVCAT)
C                           AS 40=CORFILE FOR THE DATA SET
C                           AS 50=ROSE FORMAT DATA OUT FILE
C                           AS 61-64=DEMUX DATA FILES
C
      DIMENSION MBUF(260),JCOR(140),JBUF(18),MHDR(257),SHTLN(10)
      DIMENSION C(8),COMBUF(24)
      INTEGER RCV,TYPE,SNUM,RMM,RKM,RUM,RERR,RLATD,RLATM
      INTEGER RLOND,RLONM,SLATD,SLATM,SLOND,SLONM,SDEP
      INTEGER EXPL,SIZ,BUB,SAMP,FNUM,WDEPE,WDEPI,EDEP
      INTEGER CATLFN,DEPIMX,DEPIMN,DEPEMX,DEPEMN,RDATE(3),IDAT2(3)
      INTEGER ENUMMX,ENUMMN,EXPLMX,EXPLMN,ICHNMX,HDLFN
      INTEGER ICHNMN,TYPEMX,TYPEMN,EDEPMX,EDEPMN,REP,ELOER
      INTEGER IC(24),CCODE,ENUM,IDATE(3),ITIME(3),THBUF(224),CATBUF(112)
      INTEGER CPAR(20,4),SHTLTH,TRAY1(7),TRAY(7),ISW(2),SBRK(6)
      INTEGER DBUF(54),IBUF(4032),KBUF(4096),OUTLFN,CORLFN,THLFN
      DIMENSION PLACE(5),DNAME(9),EXCODE(4)
      INTEGER F(3,12),F1(3),F2(3),F3(3),F4(3),F5(3),F6(3),F7(3)
      INTEGER F8(3),F9(2),F10(2),F11(2),F12(2),SBEG(10),SEND(10)
      REAL ILAT,ILON,ILATMX,ILATMN,ILONMX,ILONMN
      INTEGER*1 ICRAY(162),DCBUF(162)
      INTEGER*6 CSEC,SBT,DST,ISTAT,DSTMAX,DSTMIN,TDST,TDET,ISTATS(2)
      EQUIVALENCE (ISTAT,ISW),(ISW(2),ILEN)
      EQUIVALENCE (MBUF(1),ICRAY(1)),(DBUF,DCBUF)
      EQUIVALENCE (F1,F),(F2,F(1,2)),(F3,F(1,3)),(F4,F(1,4))
      EQUIVALENCE (F5,F(1,5)),(F6,F(1,6)),(F7,F(1,7)),(F8,F(1,8))
      EQUIVALENCE (F9,F(1,9)),(F10,F(1,10)),(F11,F(1,11)),(F12,F(1,12))
C
      EQUIVALENCE (CATBUF(1),JTYPE),(CATBUF(2),ITAP),(CATBUF(3),INUM)
      EQUIVALENCE (CATBUF(4),ENUM),(CATBUF(7),SBT),(CATBUF(22),WDEPI)
      EQUIVALENCE (CATBUF(5),DST),(CATBUF(9),SIZE),(CATBUF(11),RANGE)
      EQUIVALENCE (CATBUF(13),ILAT),(CATBUF(15),ILON),(CATBUF(17),ELAT)
      EQUIVALENCE (CATBUF(19),ELON),(CATBUF(21),EXPL)
      EQUIVALENCE (CATBUF(23),WDEPE),(CATBUF(24),IDEP),(CATBUF(25),EDEP)
      EQUIVALENCE (CATBUF(28),RERR),(CATBUF(29),ELOER),(CATBUF(30),BUB)
      EQUIVALENCE (CATBUF(31),SAMP),(CATBUF(32),NWDS)
      EQUIVALENCE (CATBUF(33),FNUM),(CATBUF(34),NREC),(CATBUF(35),NSAM)
      EQUIVALENCE (CATBUF(36),IDEL),(CATBUF(37),IDATE)
      EQUIVALENCE (CATBUF(27),TYPE),(CATBUF(67),SHTLIN)
      EQUIVALENCE (CATBUF(26),ICHN),(CATBUF(43),IC)
      EQUIVALENCE (CATBUF(40),IDAT2),(CATBUF(69),CCODE)
      COMMON/ITCM/TRAY,CSEC,JULD
      DATA THLFN/15/,CATLFN/30/,CORLFN/40/,OUTLFN/50/,REP/10/
      DATA HDLFN/12/,C(1)/'V'/,C(2)/'R'/,
     +C(3)/'T'/,C(4)/'P'/,C(5)/'H1'/,C(6)/'H2'/,C(7)/'W'/,C(8)/'TI'/
      DATA F1/'(1X,I8)'/,F2/'(1X,I7)'/,F3/'(1X,I6)'/,F4/'(1X,I5)'/
      DATA F5/'(1X,I4)'/,F6/'(1X,I3)'/,F7/'(1X,I2)'/,F8/'(1X,I1)'/
      DATA F9/'(I4)'/,F10/'(I3)'/,F11/'(I2)'/,F12/'(I1)'/
C
      CALL BTIME
      CALL DATE(IDATE)
      CALL TIME(ITIME)
      FOR I=1,3
      .  IDAT2(I)=IDATE(I)
      END FOR
C ZERO IC ARRAY AND SHTLN ARRAY
      FOR I=1,24
      .  IC(I)=0
      END FOR
      FOR I=1,10
      .  SHTLN(I)="      "
      END FOR
C
      WRITE(3,7950)
7950  FORMAT(1X,'Program BHIROS, REV. 4, Last modified 4/29/82')
      READ(HDLFN,8000) ITAP,IDSH,IID1,NSLN
8000  FORMAT(T50,I3,/T50,I3,/T50,I5,/T50,I3)
```

```
 81:       FOR J=1,NSLN
 82:       .  READ(HDLFN,8001) SHTLN(J),SBEG(J),SEND(J)
 83: 8001  .  FORMAT(T50,A6,2I5)
 84:       END FOR
 85:       IEVST=SBEG(1)
 86:       READ(HDLFN,8002) PLACE,IDOC,RDATE
 87: 8002  FORMAT(T50,5A6,/T50,I3,/T50,3A3)
 88:       READ(HDLFN,8003) KRA,IDSH1,IKK
 89: 8003  FORMAT(T50,I3,2(/T50,I3))
 90:       IF(IKK.NE.0)
 91:       .  READ(HDLFN,8004) KDC,IOVRD,JTC
 92: 8004  .  FORMAT(T50,I3,/T50,I4,/T50,I3)
 93:       .  IF(JTC.EQ.1)
 94:       .  .  READ(HDLFN,8022) TDST,TDET
 95: 8022  .  .  FORMAT(2I14)
 96:       .  END IF
 97:       END IF
 98: C ENCODE HIG HEADER INFO INTO TAPE HEAD BUFFER
 99: C
100:       JTYPE=2
101:       ENCODE(84,7500,THBUF) JTYPE,ITAP,NSLN,(PLACE(I),I=1,5),IDOC,
102:      +(RDATE(I),I=1,3),(IDATE(I),I=1,3),(IDATE(I),I=1,3),
103:      +(IDATE(I),I=1,3)
104: 7500  FORMAT(I3,2I6,5A6,13A3)
105:       ENCODE(3,7501,THBUF(160)) IDSH
106: 7501  FORMAT(I3)
107: C READ,WRITE TAPE HEADER BUILT WITH PROGRAM ROSEHD
108:       OPEN THLFN
109:       OPEN OUTLFN
110:       LOOP
111:       .  BUFFER IN (THLFN,MBUF,B,256,MSTAT,MLEN)
112:       .  CALL STATUS(THLFN)
113:       .  EXIT LOOP IF (MSTAT.GE.3)
114:       .  L=0
115:       .  FOR I=1,4
116:       .  .  FOR J=1,20
117:       .  .  .  CPAR(J,I)=MBUF(54+J+L)
118:       .  .  END FOR
119:       .  .  L=L+20
120:       .  END FOR
121:       .  FOR I=55,260
122:       .  .  MBUF(I)=0
123:       .  END FOR
124:       .  FOR I=1,4
125:       .  .  IC(I)=CPAR(1,I)
126:       .  END FOR
127:       .  BUFFER OUT(OUTLFN,MBUF,B,256,MSTAT,MLEN)
128:       .  CALL STATUS(OUTLFN)
129:       .  ENDFILE OUTLFN
130:       .  WRITE(3,200)
131: 200   .  FORMAT(' TAPE HEADER WRITTEN')
132: C DECODE TAPE HEADER FOR NEEDED INFO
133:       .  J=1
134:       .  FOR I=1,162
135:       .  .  K=MOD(I,3)
136:       .  .  IF(K.NE.1)
137:       .  .  .  DCBUF(J)=ICRAY(I)
138:       .  .  .  J=J+1
139:       .  .  END IF
140:       .  END FOR
141: C
142: C DECODE IID
143:       .  J=1
144:       .  I=0
145:       .  FOR K=4,1,-1
146:       .  .  IF(DCBUF(K).NE.' ')
147:       .  .  .  I=I+1
148:       .  .  .  IF(I.EQ.1)
149:       .  .  .  .  IP=J+8
150:       .  .  .  END IF
151:       .  .  END IF
152:       .  .  J=J+1
153:       .  END FOR
154:       .  DECODE(4,F(1,IP),DBUF) IID
155: C DECODE NUMBER OF FILES
156:       .  J=1
157:       .  I=0
158:       .  FOR K=108,101,-1
159:       .  .  IF(DCBUF(K).NE.' ')
160:       .  .  .  I=I+1
```

```
161:        .  .  .  IF(I.EQ.1)
162:        .  .  .  .  IP=J
163:        .  .  .  END IF
164:        .  .  END IF
165:        .  .  J=J+1
166:        .  END FOR
167:        .  DECODE(9,F(1,IP),DBUF(34)) NFILES
168: C DECODE THE REST
169:        .  DECODE(55,8005,DBUF(2)) DNAME
170: 8005   .  FORMAT(1X,9A6)
171:        .  DECODE(20,8006,DBUF(21)) EXCODE
172: 8006   .  FORMAT(3A6,A2)
173:        .  DECODE(12,8010,DBUF(27))ISYR,ISMO,ISDA,ISHR,ISMIN
174:  8010  .  FORMAT(2X,5I2)
175:        .  DECODE(10,8020,DBUF(31))IFYR,IFMO,IFDA,IFHR,IFMIN
176:  8020  .  FORMAT (5I2)
177:        .  WRITE(3,8040) IID,NFILES
178:  8040  .  FORMAT(1X,"INSTRUMENT ID  ",I4,/," # FILES  ",I4)
179:        .  IF(IOVRD.NE.0)
180:        .  .  NFILES=IOVRD
181:        .  .  WRITE(3,8041) NFILES
182: 8041   .  .  FORMAT(' # FILES RESET TO:',I6)
183:        .  END IF
184:        .  WRITE(3,1120) IEVST
185: 1120   .  FORMAT(1X,'STARTING WITH EVENT #:',I5)
186:        .  WRITE(3,8045) DNAME
187: 8045   .  FORMAT(1X,'DESIGNER NAME & ADDRESS: ',9A6)
188:        .  WRITE(3,8046) EXCODE
189: 8046   .  FORMAT(1X,3A6,A2,' EXPERIMENT')
190:        .  WRITE(3,8050)ISYR,ISMO,ISDA,ISHR,ISMIN
191: 8050   .  FORMAT(1X,"START TIME  ",5(1X,I2))
192:        .  WRITE(3,8060)IFYR,IFMO,IFDA,IFHR,IFMIN
193: 8060   .  FORMAT(1X,"END TIME    ",5(1X,I2))
194:        .  IF(JTC.EQ.0)
195:        .  .  TRAY(1)=ISYR
196:        .  .  TRAY(2)=ISMO
197:        .  .  TRAY(3)=ISDA
198:        .  .  TRAY(4)=ISHR
199:        .  .  TRAY(5)=ISMIN
200:        .  .  TRAY(6)=0
201:        .  .  TRAY(7)=0
202:        .  .  CALL ITMCNT
203:        .  .  TDST=CSEC
204:        .  .  TRAY(1)=IFYR
205:        .  .  TRAY(2)=IFMO
206:        .  .  TRAY(3)=IFDA
207:        .  .  TRAY(4)=IFHR
208:        .  .  TRAY(5)=IFMIN
209:        .  .  CALL ITMCNT
210:        .  .  TDET=CSEC
211:        .  END IF
212: C ENCODE TAPE HEADER INFO INTO CURRENT TAPE BUFFER
213: C
214:        .  ENCODE(174,7600,THBUF(95)) IID,DNAME,EXCODE,NFILES,TDST,TDET,SHTLN
215: 7600   .  FORMAT(I4,12A6,A2,I6,I14,1X,I14,1X,10A6)
216: C
217:        END LOOP
218:        FOR I=1,260
219:        .  MBUF(I)=0
220:        END FOR
221: C
222: C GET HEADER INFO FROM DEMUX FILE PLUS POSITION TO GET MORE LATER
223:        FOR I=1,4
224:        .  LFN=60+I
225:        .  OPEN LFN
226:        END FOR
227:        LFN=61
228:        CALL DPOS(LFN,2)
229:        CALL BUFIN(LFN,MBUF,112,IEOF)
230: C COMPUTE # SAMPLES IN LAST RECORD FOR EACH SHOT FROM SHOT LENGTH
231:        SHTLTH=MBUF(111)
232:        NSAM=MOD(SHTLTH,4096)
233:        IF(NSAM.NE.0)
234:        .  NREC=SHTLTH/4096+1
235:        ELSE
236:        .  NREC=SHTLTH/4096
237:        END IF
238: D      WRITE(3,5005) SHTLTH,NREC,NSAM
239: D5005  FORMAT(1X,'SHOT LTH:',I6,'NO. RECS/SHT:',I4,' REMAINDER:',I5)
240:        FOR I=1,4
```

```
241:       . CPAR(11,I)=NREC
242:       . CPAR(12,I)=NSAM
243:       END FOR
244: C POSITION DEMUX FILE #1(LFN 61) TO ITS HEADERS
245:       CALL DPOS(LFN,MBUF(112))
246: D     WRITE(3,201)
247: D201  FORMAT(1X,'DEMUX FILE POSITIONED')
248: C
249:       IDEL=0
250: C ZERO WORKING BUFFER
251:       FOR I=1,260
252:       . MBUF(I)=0
253:       END FOR
254: C FIND POSITION OF START OF TAPE HEADERS IN CATALOG
255:       OPEN CATLFN
256:       CALL DPOS(CATLFN,2)
257:       CALL BUFIN(CATLFN,MBUF,112,IEOF)
258:       ITHEAD=MBUF(112)
259: C SAVE OFF TAPE HEADER RECORDS
260: C
261:       OPEN 11
262:       INEXT=ITHEAD
263:       CALL DPOS(CATLFN,INEXT)
264:       LOOP
265:       . ISAV=0
266:       . CALL BUFIN(CATLFN,MBUF,224,IEOF)
267:       . EXIT LOOP IF(IEOF.GE.3)
268: C IF REARCHIVING, DONT SAVE TAPE HEADER RECORD FOR THIS TAPE
269:       . IF(KRA.EQ.1)
270:       . . DECODE(9,7550,MBUF) KTAP
271: 7550  . . FORMAT(3X,I6)
272:       . . DECODE(6,7553,MBUF(95)) INM
273: 7553  . . FORMAT(I4,2X)
274:       . . DECODE(6,7551,MBUF(121)) NF
275: 7551  . . FORMAT(I6)
276:       . . DECODE(3,7501,MBUF(160)) IDASH
277:       . . IF(KTAP.EQ.ITAP)
278:       . . . IF(IDSH1.EQ.IDASH.AND.NFILES.EQ.NF.AND.IID.EQ.INM)
279:       . . . . WRITE(3,7552) ITAP,IDSH1,IID,NFILES,KTAP,IDASH,INM,NF
280: 7552  . . . . FORMAT(1X,' Input tape,instrument,#files:',I6,'-',I2,2I6,
281:      +. . . . /'   catalog file tape,instrument,#files:',I6,'-',I2,2I6)
282:       . . . . WRITE(3,7554)
283: 7554  . . . . FORMAT(' We have a match;Execute replacement ARCHIVE',
284:      +. . . . ' after saving remaining headers')
285:       . . . . INEXT=MBUF(153)
286:       . . . . ISAV=1
287:       . . . END IF
288:       . . END IF
289:       . END IF
290:       . IF(ISAV.EQ.0)
291:       . . BUFFER OUT(11,MBUF,B,224,MSTAT,MLEN)
292:       . . CALL STATUS(11)
293:       . END IF
294:       END LOOP
295:       REWIND 11
296:       THBUF(153)=INEXT
297:       CALL DPOS(CATLFN,INEXT)
298:       WRITE(3,1050)
299: 1050  FORMAT(/1X,"Begin reading & writing event header records")
300: C ***************************************************************
301: C  LOOP TO BUILD ARCHIVE FILE HEADER RECORDS
302: C  (BOTH FOR ARCHIVE FORMAT AND CATALOG FILE)
303: C ***************************************************************
304: C            <<<<<<< FIRST CORFILE >>>>>>>>                @@@@@@@@@@@
305:       FNUM=1
306:       JTYPE=1
307:       OPEN 20
308:       OPEN CORLFN
309:       LOOP
310: 11    . BUFFER IN (CORLFN,JCOR,B,140,JSTAT,JLEN)
311:       . CALL STATUS(CORLFN)
312:       . EXIT LOOP IF(JSTAT.GE.3)
313:       . DECODE(60,7011,JCOR) RCV,SNUM,(SBRK(I),I=1,6),K
314: 7011  . FORMAT(2I6,2X,I4,I3,3I2,I3,I2,28X)
315:       . IF(FNUM.EQ.1)
316:       . . IF(SNUM.NE.IEVST) GO TO 11
317:       . END IF
318:       . WRITE(3,7011) RCV,SNUM,(SBRK(I),I=1,6),K
319:       . DECODE(60,7012,JCOR(21)) SCORR,SD,SIZE,RCORR,RD,RANGE
320: 7012  . FORMAT(6F10.4)
```

```
321:       .  DECODE(60,7012,JCOR(41)) CSRAN,BTC,SBATC,SDC,SDIST,RBATH
322:       .  DECODE(60,7013,JCOR(61)) RDC,RDIST,RV,WV1,ELAT,BUB,EXPL
323: 7013  .  FORMAT(5F10.4,I4,I2,4X)
324:       .  DECODE(60,7014,JCOR(81)) ELON,BATHS,BLS,ILAT
325: 7014  .  FORMAT(10X,4F10.4,10X)
326:       .  DECODE(60,7015,JCOR(101)) ILON,TT,TDIST,TPS1,TPS2
327: 7015  .  FORMAT(10X,5F10.4)
328:       .  DECODE(60,7016,JCOR(121)) TPS3,CTDAT,CTDATR,WWC1,WWC2,SRKM
329: 7016  .  FORMAT(6F10.4)
330: C CONVERT TO ROSE FORMAT
331:       .  RMM=IFIX(RANGE*1000.)/1000000
332:       .  RKM=IFIX(RANGE)-RMM*1000
333:       .  RUM=RUP(RANGE*1000.)-RKM*1000-RMM*1000000
334:       .  RERR=500
335:       .  RLATD=IFIX(ILAT)
336:       .  RLATM=(RUP(ILAT*1000.)-RLATD*1000)
337:       .  RLOND=IFIX(ILON)
338:       .  RLONM=(RUP(ILON*1000.)-RLOND*1000)
339:       .  SLATD=IFIX(ELAT)
340:       .  SLATM=(RUP(ELAT*1000.)-SLATD*1000)
341:       .  SLOND=IFIX(ELON)
342:       .  SLONM=(RUP(ELON*1000.)-SLOND*1000)
343:       .  RDM=RD*1000.
344:       .  IDEP=RUP(RDM)
345:       .  RD=(RD/1.5)*1000.
346:       .  WDEPI=RUP(RD)
347:       .  BATHS=(BATHS/2.0)*1.5
348:       .  WDEPE=RUP(BATHS)
349:       .  ELOER=250
350:       .  TYPE=2
351:       .  IF(SBRK(1).LT.1900)
352:       .  .  SBRK(1)=SBRK(1)+1900
353:       .  END IF
354:       .  TRAY(1)=SBRK(1)
355:       .  JULD=SBRK(2)
356:       .  FOR I=3,6
357:       .  .  TRAY(I+1)=SBRK(I)
358:       .  END FOR
359:       .  CALL LJLCNT
360: C CORRECT SHOT BREAK TIME WITH BURN TIME CORRECTION
361:       .  IBTC=0
362:       .  IF(BTC.NE.0.0)
363:       .  .  IBTC=RUP(BTC*1000.)
364:       .  .  CSEC=CSEC-IBTC
365:       .  END IF
366:       .  CALL CNTITM
367:       .   FOR I=1,7
368:       .  .  MHDR(I+27)=TRAY(I)
369:       .  END FOR
370:       .  SIZ=RUP((ALOG10(SIZE*1000.))*1000.)
371:       .  EDEP=SD*1000.
372:       .  SDEP=RUP(SD*1000.)
373:       .  RERR=500
374:       .  ICHN=4
375:       .  NWDS=4096
376:       .  CCODE=0
377:       .  MHDR(60)=CCODE
378:       .  FOR I=1,4
379:       .  .  L=10*(6+2*(I-1))
380:       .  .  FOR K=1,20
381:       .  .  .  LL=L+K
382:       .  .  .  MHDR(LL)=CPAR(K,I)
383:       .  .  END FOR
384:       .  END FOR
385:       .  FOR I=42,59
386:       .  .  MHDR(I)=0
387:       .  END FOR
388:       .  FOR I=141,256
389:       .  .  MHDR(I)=0
390:       .  END FOR
391: C            <<<<<<<< SECOND DEMUX HEADERS >>>>>>>>>>>
392: 12    .  CALL BUFIN(LFN,JBUF,18,IEOF)
393:       .  EXIT LOOP IF(IEOF.GE.3)
394:       .  DECODE(54,7000,JBUF)INUM,ENUM,ICH,SBT,DST,SAMP,IDA
395: 7000  .  FORMAT(2I6,I2,2I16,I3,2X,A3)
396:       .  IF(FNUM.EQ.1)
397:       .  .  IF(ENUM.NE.IEVST) GO TO 12
398:       .  END IF
399:       .  SBT=SBT-IBTC
400:       .  JTCK=CSEC-SBT
```

```
401:       .  IF(JTCK.NE.0)
402:       .  .  WRITE(3,5052) JTCK,ENUM
403: 5052  .  .  FORMAT(1X,'SHOT BREAK TIME ON DEMUX FILE DOESNT MATCH CORFILE'
404:      +.  .  /'  TWO TIMES DIFFER BY ',I6,'  MSEC FOR SHOT',I6)
405:       .  END IF
406:       .  MHDR(257)=IDA
407:       .  CSEC=DST
408:       .  CALL CNTITM
409:       .  FOR I=1,7
410:       .  .  MHDR(I+3)=TRAY(I)
411:       .  END FOR
412: D     CSEC=SBT
413: D     CALL CNTITM
414: D     WRITE(3,5050)INUM,ENUM,TRAY,(MHDR(I),I=4,10)
415: D5050 FORMAT(' R= ',I4,' S=',I5,' SBT&DST: ',T30,I4,5I3,I4,T55,I4,
416: D    .5I3,I4)
417: D       WRITE(3,5057) IDA
418: D5057   FORMAT(1X,'**********DATA START ADDR: ',I6)
419: C COMPARE DEMUX & CORFILE FOR RCV/SHT MATCH
420:       .  IF (INUM.NE.RCV)
421:       .  .  WRITE(3,3040) INUM,RCV
422: 3040  .  .  FORMAT(' RCV#S DEMUX VS CORFILE DO NOT MATCH:',2I6)
423:       .  .  IERR=1
424:       .  .  EXIT LOOP
425:       .  END IF
426:       .  IF (ENUM.NE.SNUM)
427:       .  .  WRITE(3,3050) ENUM,SNUM
428: 3050  .  .  FORMAT(' SHOT#S DEMUX VS CORFILE DO NOT MATCH:',2I6)
429:       .  .  IERR=1
430:       .  .  EXIT LOOP
431:       .  END IF
432:       .  FOR JK=1,NSLN
433:       .  .  IF(ENUM.GE.SBEG(JK).AND.ENUM.LE.SEND(JK))
434:       .  .  .  SHTLIN=SHTLN(JK)
435:       .  .  END IF
436:       .  END FOR
437: C WRITE VARIABLES, INCLUDING KEYWORDS,  TO CATALOG
438: C
439:       .  CALL BUFOUT(CATLFN,CATBUF,112,IEOF)
440:       .  CALL DSTAT(CATLFN,ISTATS,ICRA)
441: C STORE ROSE FORMAT FILE HEADERS IN TEMP FILE
442: C NOTE* WORD 257 IS IDA, THE DATA ADDRESS FOR EACH EVENT
443:       .  ENCODE(9,5501,MHDR) RCV,TYPE,SNUM
444: 5501  .  FORMAT(3A3)
445:       .  ENCODE(51,5502,MHDR(11)) RMM,RKM,RUM,RERR,RLATD,RLATM,RLOND,
446:      +.  RLONM,IDEP,WDEPI,SLATD,SLATM,SLOND,SLONM,ELOER,SDEP,WDEPE
447: 5502  .  FORMAT(17A3)
448:       .  ENCODE(21,5503,MHDR(35)) EXPL,SIZ,BUB,SAMP,ICHN,NWDS,FNUM
449: 5503  .  FORMAT(7A3)
450:       .  BUFFER OUT(20,MHDR,B,257,MSTAT,MLEN)
451:       .  CALL STATUS(20)
452: C INITIALIZE MIN,MAX VALUES OF KEYWORDS
453: C
454:       .  IF(FNUM.EQ.1)
455:       .  .  INUMMX=INUM
456:       .  .  INUMMN=INUM
457:       .  .  ENUMMX=ENUM
458:       .  .  ENUMMN=ENUM
459:       .  .  DSTMAX=DST
460:       .  .  DSTMIN=DST
461:       .  .  EXPLMX=EXPL
462:       .  .  EXPLMN=EXPL
463:       .  .  DEPIMX=WDEPI
464:       .  .  DEPIMN=WDEPI
465:       .  .  DEPEMX=WDEPE
466:       .  .  DEPEMN=WDEPE
467:       .  .  IDEPMX=IDEP
468:       .  .  IDEPMN=IDEP
469:       .  .  EDEPMX=EDEP
470:       .  .  EDEPMN=EDEP
471:       .  .  ICHNMX=ICHN
472:       .  .  ICHNMN=ICHN
473:       .  .  TYPEMX=TYPE
474:       .  .  TYPEMN=TYPE
475:       .  .  SIZEMX=SIZE
476:       .  .  SIZEMN=SIZE
477:       .  .  RANGMX=RANGE
478:       .  .  RANGMN=RANGE
479:       .  .  ILATMX=ILAT
480:       .  .  ILATMN=ILAT
```

```
481:        . .  ILONMX=ILON
482:        . .  ILONMN=ILON
483:        . .  ELATMX=ELAT
484:        . .  ELATMN=ELAT
485:        . .  ELONMX=ELON
486:        . .  ELONMN=ELON
487:        .  END IF
488: C NOW COMPARE NEW VALUES OF KEYWORDS WITH MIN,MAX AND GET NEW MIN,MAX
489: C
490:        .  INUMMX=MAX0(INUMMX,INUM)
491:        .  INUMMN=MIN0(INUMMN,INUM)
492:        .  ENUMMX=MAX0(ENUMMX,ENUM)
493:        .  ENUMMN=MIN0(ENUMMN,ENUM)
494:        .  DSTMAX=MAX2(DSTMAX,DST)
495:        .  DSTMIN=MIN2(DSTMIN,DST)
496:        .  EXPLMX=MAX0(EXPLMX,EXPL)
497:        .  EXPLMN=MIN0(EXPLMN,EXPL)
498:        .  DEPIMX=MAX0(DEPIMX,WDEPI)
499:        .  DEPIMN=MIN0(DEPIMN,WDEPI)
500:        .  DEPEMX=MAX0(DEPEMX,WDEPE)
501:        .  DEPEMN=MIN0(DEPEMN,WDEPE)
502:        .  IDEPMX=MAX0(IDEPMX,IDEP)
503:        .  IDEPMN=MIN0(IDEPMN,IDEP)
504:        .  EDEPMX=MAX0(EDEPMX,EDEP)
505:        .  EDEPMN=MIN0(EDEPMN,EDEP)
506:        .  ICHNMX=MAX0(ICHNMX,ICHN)
507:        .  ICHNMN=MIN0(ICHNMN,ICHN)
508:        .  TYPEMX=MAX0(TYPEMX,TYPE)
509:        .  TYPEMN=MIN0(TYPEMN,TYPE)
510:        .  SIZEMX=AMAX1(SIZEMX,SIZE)
511:        .  SIZEMN=AMIN1(SIZEMN,SIZE)
512:        .  RANGMX=AMAX1(RANGMX,RANGE)
513:        .  RANGMN=AMIN1(RANGMN,RANGE)
514:        .  ILATMX=AMAX1(ILATMX,ILAT)
515:        .  ILATMN=AMIN1(ILATMN,ILAT)
516:        .  ILONMX=AMAX1(ILONMX,ILON)
517:        .  ILONMN=AMIN1(ILONMN,ILON)
518:        .  ELATMX=AMAX1(ELATMX,ELAT)
519:        .  ELATMN=AMIN1(ELATMN,ELAT)
520:        .  ELONMX=AMAX1(ELONMX,ELON)
521:        .  ELONMN=AMIN1(ELONMN,ELON)
522:        .  N=ICHN*NREC
523: D       WRITE(3,4112) ENUM
524: D4112  FORMAT(1X,'Processing header for event no.  ',I6)
525:        .  FNUM=FNUM+1
526:        .  EXIT LOOP IF (FNUM.GT.NFILES)
527:       END LOOP
528:       IF(IERR.EQ.1) STOP ERR
529: C*************************************************************************
530:       WRITE(3,4113)
531:  4113 FORMAT(/1X,'Finished processing event header records')
532: C CHECK FOR INSTRUMENT KEYWORD VALUE MISMATCHES
533:       IF(ILATMX.NE.ILATMN)
534:       .  WRITE(3,2000)
535: 2000  .  FORMAT(1X,'INST. LATITUDES ENCODED NOT EQUAL FOR ALL EVENTS')
536:       END IF
537:       IF(INUMMN.NE.INUMMX)
538:       .  WRITE(3,2001)
539: 2001  .  FORMAT(1X,'INST # ENCODED NOT EQUAL FOR ALL EVENTS')
540:       END IF
541:       IF(DEPIMN.NE.DEPIMX)
542:       .  WRITE(3,2002)
543: 2002  .  FORMAT(1X,'WATER DEPTHS AT INST NOT SAME FOR ALL EVENTS')
544:       END IF
545:       IF(IDEPMN.NE.IDEPMX)
546:       .  WRITE(3,2003)
547: 2003  .  FORMAT(1X,'INST. DEPTHS NOT SAME FOR ALL EVENTS')
548:       END IF
549: C CHECK FOR NEGATIVE LATS AND LONS; REVERSE THEM
550: C
551:       IF(ELATMX.LT.0.0.AND.ELATMN.LT.0.0)
552:       .  TEMP=ELATMX
553:       .  ELATMX=ELATMN
554:       .  ELATMN=TEMP
555:       END IF
556:       IF(ELONMX.LT.0.0.AND.ELONMN.LT.0.0)
557:       .  TEMP=ELONMX
558:       .  ELONMX=ELONMN
559:       .  ELONMN=TEMP
560:       END IF
```

```
561: C ENCODE MIN,MAX KEYWORD VALUES INTO THBUF
562: C
563:       ENCODE(96,6000,THBUF(29)) INUMMN,ENUMMX,ENUMMN,DSTMAX,
564:      +DSTMIN,EXPLMX,EXPLMN,DEPIMN,DEPEMX,DEPEMN,
565:      +IDEPMN,EDEPMX,EDEPMN,ICHNMX,ICHNMN,TYPEMX,TYPEMN
566: 6000  FORMAT(3I6,2I13,2I2,6I6,2I3,2I2,2X)
567:       ENCODE(102,6001,THBUF(61)) SIZEMX,SIZEMN,RANGMX,RANGMN,
568:      +ILATMN,ILONMN,ELATMX,ELATMN,ELONMX,ELONMN
569: 6001  FORMAT(10F10.4,2X)
570:       REWIND 20
571:       REWIND LFN
572: D     WRITE(3,202)
573: D202  FORMAT(1X,'NOW READY TO PUT OUT ROSE FORMAT HEADERS+DATA')
574: C
575: C ***************************************************************************
576: C NOW BUILD ROSE FORMAT DATA FILE WITH HEADERS
577: C ***************************************************************************
578: C
579: C READ/WRITE FILE HEADER RECORDS AND DATA
580: C
581:       LOOP(NFILES)
582:       .  BUFFER IN(20,MHDR,B,257,MSTAT,MLEN)
583:       .  CALL STATUS(20)
584:       .  IDA=MHDR(257)
585:       .  WRITE(3,101) MHDR(3)
586: 101   .  FORMAT(1X,'Processing data for event #: ',I6)
587:       .  EXIT LOOP IF(MSTAT.GE.3)
588:       .  BUFFER OUT(OUTLFN,MHDR,B,256,MSTAT,MLEN)
589:       .  CALL STATUS(OUTLFN)
590:       .  FOR I=1,4
591:       .  .  K1EOF=2
592:       .  .  KEOF=2
593:       .  .  LFN=60+I
594:       .  .  IPTR=1
595:       .  .  ILEN=0
596:       .  .  CALL DPOS(LFN,IDA)
597:       .  .  LOOP
598:       .  .  .  FOR J=1,4096
599:       .  .  .  .  IF(IPTR.GT.ILEN)
600:       .  .  .  .  .  K1EOF=KEOF
601:       .  .  .  .  .  EXIT FOR IF(KEOF.GE.3)
602:       .  .  .  .  .  CALL BUFIN(LFN,IBUF,4032,KEOF)
603:       .  .  .  .  .  CALL DSTAT(LFN,ISTAT,ICRA)
604:       .  .  .  .  .  EXIT FOR IF(ILEN.EQ.0)
605:       .  .  .  .  .  IPTR=1
606:       .  .  .  .  END IF
607:       .  .  .  .  KBUF(J)=IBUF(IPTR)
608:       .  .  .  .  IPTR=IPTR+1
609:       .  .  .  END FOR
610:       .  .  .  FOR K=J,4096
611:       .  .  .  .  KBUF(K)=0
612:       .  .  .  END FOR
613:       .  .  .  CALL BUFOUT(OUTLFN,KBUF,4096,JEOF)
614:       .  .  .  J=J-1
615:       .  .  .  ITOTAL=ITOTAL+J
616:       .  .  .  EXIT LOOP IF(K1EOF.GE.3)
617:       .  .  END LOOP
618:       .  END FOR
619:       .  ENDFILE OUTLFN
620:       END LOOP
621: C CLOSE FILES EXCEPT FOR CATALOG FILE
622:       FOR I=1,4
623:       .  LFN=60+I
624:       .  CLOSE LFN
625:       END FOR
626:       CLOSE 20
627:       CLOSE CORLFN
628:       CLOSE OUTLFN
629:       WRITE(3,4000) MHDR(3),MHDR(1),MHDR(39),LFN
630: 4000  FORMAT(' LAST SHOT WRITTEN WAS: ',I6,' FOR RCV # ',
631:      .I6, ' CHAN# ',I3,' LFN ',I2)
632: C ADD TAPE HEADER RECORDS BACK ON TO EVENT CATALOG
633: C
634:       CALL DSTAT(CATLFN,ISTATS,ICRA)
635:         IF(KRA.EQ.1)
636:       .    CALL DPOS(CATLFN,ITHEAD)
637:         END IF
638:       LOOP
639:       .  BUFFER IN(11,MBUF,B,224,MSTAT,MLEN)
640:       .  CALL STATUS(11)
```

```
641:        .  EXIT LOOP IF(MSTAT.GE.3)
642:        .  CALL BUFOUT(CATLFN,MBUF,224,IEOF)
643:        END LOOP
644:        CALL DSTAT(CATLFN,ISTATS,ILAST)
645:          IF(KRA.NE.1)
646:        .  CALL DPOS(CATLFN,2)
647:        .  CALL BUFIN(CATLFN,MBUF,112,IEOF)
648:        .  MBUF(112)=ICRA
649:        .  CALL DPOS(CATLFN,2)
650:        .  CALL BUFOUT(CATLFN,MBUF,112,IEOF)
651:        .  WRITE(3,1121) ICRA
652: 1121   .  FORMAT(1X,'START RECORD FOR TAPE HEADERS NOW:',I6)
653:        END IF
654: C WRITE CURRENT TAPE HEADER TO CATALOG FILE AND CLOSE IT
655: C ALSO WRITE CURRENT TAPE HEADER TO THEAD
656:        CALL DPOS(CATLFN,ILAST)
657:        CALL BUFOUT(CATLFN,THBUF,224,IEOF)
658:        BUFFER OUT(11,THBUF,B,224,MSTAT,MLEN)
659:        ENDFILE 11
660:        ENDFILE CATLFN
661:        CLOSE CATLFN
662: C ***************************************************************
663: C PRINTED REPORT
664: C ***************************************************************
665:        OPEN REP
666:        IPG=1
667:        WRITE(REP,1) IPG
668: 1      FORMAT('1'//T53,'ROSE ARCHIVE REPORT',T114,'PAGE NO. ',I3)
669:        WRITE(REP,2)
670: 2      FORMAT(//T45,'* * * SUMMARY OF DATA ARCHIVED * * *' ,
671:       .T112,'RUN DATE,TIME')
672:          IF(KRA.NE.1)
673:        .  WRITE(REP,3) IDATE,ITIME
674: 3      .  FORMAT(T111,3A3,', ',3A3)
675:          ELSE
676:        .    WRITE(REP,4) IDATE,ITIME
677: 4      .    FORMAT(T52,'<<REPLACEMENT ARCHIVE>>',T111,3A3,', ',3A3)
678:          END IF
679: C ENCODE COMPONENT TYPES INTO COMBUF
680:        ALPHA='  '
681:        FOR K=1,24
682:        .  ENCODE(3,6777,COMBUF(K)) ALPHA
683:        END FOR
684:        FOR K=1,ICHN
685:        .  FOR J=1,8
686:        .  .  IF(IC(K).EQ.J)
687:        .  .  .  ENCODE(3,6777,COMBUF(K)) C(J)
688: 6777   .  .  .  FORMAT(A3)
689:        .  .  END IF
690:        .  END FOR
691:        END FOR
692: C TRANSLATE START TIMES FOR PRINTING
693:        CSEC=DSTMIN
694:        CALL CNTITM
695:        FOR I=1,5
696:        .  TRAY1(I)=TRAY(I)
697:        END FOR
698:        CSEC=DSTMAX
699:        CALL CNTITM
700:        WRITE(REP,400) ITAP,IDSH,RDATE,EXCODE
701: 400    FORMAT(//T5,'RARC TAPE # ',I5,'-',I3,T25,'DATE RECEIVED: ',
702:       +3A3,T82,'EXPERIMENT: ',3A6,A2)
703:        WRITE(REP,401) ISYR,ISMO,ISDA,ISHR,ISMIN
704: 401    FORMAT(T5,'TAPE DATA START TIME: ',5I4)
705:        WRITE(REP,402) IFYR,IFMO,IFDA,IFHR,IFMIN
706: 402    FORMAT(T5,'TAPE DATA END TIME  : ',5I4)
707:        WRITE(REP,403) IID,DNAME
708: 403    FORMAT(/T5,'INSTRUMENT #: ',I4,T48,
709:       +'DESIGNER: ',9A6)
710:        WRITE(REP,404) IDOC,PLACE
711: 404    FORMAT(T5,'DOCUMENTATION CODE (YES=1): ',A3,T48,
712:       +'INSTITUTION RECEIVED FROM: ',5A6)
713:        WRITE(REP,405) (COMBUF(I),I=1,10),ILAT,ILON
714: 405    FORMAT(T5,'COMPONENTS 1-10 ONLY: ',10A2,T48,'INSTRUMENT LATITUDE:'
715:       +F8.4,T80,'INSTRUMENT LONGITUDE: ',F10.4)
716:        WRITE(REP,406) IDEP,WDEPI
717: 406    FORMAT(T5,'INSTRUMENT DEPTH: ',I6,' M.',T48,
718:       +'WATER DEPTH AT INSTRUMENT: ',I4,' MSEC.')
719:        WRITE(REP,407) NFILES
720: 407    FORMAT(T5,'NUMBER OF EVENTS: ',I4)
```

```
      WRITE(REP,408)
408   FORMAT(//T5,'EVENT KEYWORD MINIMUM & MAXIMUM VALUES:')
      IF(NSLN.EQ.1)
      . WRITE(REP,304) TYPEMX,SHTLN(1)
304   . FORMAT(/T20,'EVENT TYPE  ',I2,T48,'SHOT LINE #: ',A6)
      END IF
      IF(NSLN.GT.1)
      . WRITE(REP,325) TYPEMX,SHTLN
325   . FORMAT(/T20,'EVENT TYPE ',I2,T48,'SHOT LINE #S: ',
     +. A6,9(' ',A6))
      END IF
      WRITE(REP,305) ENUMMN,ENUMMX
305   FORMAT(T20,'EVENT #S  ',T52,I6,T60,' TO ',I6)
      WRITE(REP,306) (TRAY1(I),I=1,5),(TRAY(I),I=1,5)
306   FORMAT(T20,'DATA START TIMES  ', 5I4,T60,' TO ',I6,4I4)
      WRITE(REP,308) ELATMN,ELATMX
308   FORMAT(T20,'EVENT LATITUDES  ',T48,F10.3,T60,' TO ',F10.3)
      WRITE(REP,309) ELONMN,ELONMX
309   FORMAT(T20,'EVENT LONGITUDES ',T48,F10.3,T60,' TO ',F10.3)
      WRITE(REP,310) EDEPMN,EDEPMX
310   FORMAT(T20,'EVENT DEPTHS  ',T53,I5,T60,' TO  ',I5,' M')
      WRITE(REP,311) DEPEMN,DEPEMX
311   FORMAT(T20,'WATER DEPTHS  ',T52,I6,T60,' TO ',I6,' M')
      WRITE(REP,312) SIZEMN,SIZEMX
312   FORMAT(T20,'EVENT SIZES  ',T53,F5.1,T60,' TO  ',F5.1,' KG')
      IF(EXPLMN.EQ.EXPLMX) GO TO 10
      WRITE(REP,313) EXPLMN,EXPLMX
313   FORMAT(T20,'EXPLOSIVE TYPES  ',T56,I2,T60,' TO ',I2)
      GO TO 15
10    WRITE(REP,316) EXPLMN
316   FORMAT(T20,'EXPLOSIVE TYPE  ',T48,I2)
15    WRITE(REP,314) RANGMN,RANGMX
314   FORMAT(T20,'RANGES-EVENT TO RCVR ',T52,F6.1,T59,' TO ',F6.1,' KM')
      WRITE(REP,317)
317   FORMAT(1H1)
      CALL ETIME
      STOP YEA
999     STOP EOF
      END
      FUNCTION RUP(X)
      J=IFIX(X)
      C=FLOAT(J)
      B=(X-C)*10.
      IF(B.GE.5)
      . RUP=J+1
      ELSE
      . RUP=J
      END IF
      RETURN
      END
```

```
1:  $MS
2:  RW 3
3:  MO RE
4:  $PR MACRO H R A M A C -  To archive HIG formatted demux data with
5:  $PR                      corfile. Outputs to Catalog file and to
6:  $PR                      an OUTFILE specified by user. Also will
7:  $PR                      write archived data to the archive tape by
8:  $PR                      specifying .C option (HRAMAC.C).
9:  IF,(.NOT.(C.SPA.A&0)) $JU !GEN
10: $PR ********* YOU HAVE SELECTED COPY OPTION ***************************
11: $PR ENTER NAME OF HEADER FILE (File containing HIG ROSE tape header(s))
12: $SR.IT #FIL
13: $AS 11=#FIL
14: $PR TAPE DRIVE 12
15: /PS 12
16: PR TAPE DRIVE 10
17: /PS 10
18: PR TAPE DRIVE 9
19: /PS 9
20: PR ENTER DRIVE NO. TO RESOURCE
21: $SR.IN #NUM
22: SR.N #TAP=1
23: $JU !COPY
24: !GEN PR ********* YOU HAVE SELECTED ARCHIVE OPTION ******************
25: PR
26: $PR Did you build header files? If not, ABORT & run M<BUILD
27: PR
28: $PR ENTER NAME FOR DATA OUTFILE
29: $SR.IT #FIL
30: $GE #FIL G500 M100000 P3
31: AS 50=#FIL
32: JE 318 !BIG
33: PR ENTER NAME OF HIG HEADER FILE (H+INST NO)
34: SR.IT #HED
35: GE REPORT P3
36: AS 10=REPORT
37: GE THEAD P1
38: AS 11=THEAD
39: AS 12=#HED
40: PR ENTER NAME OF ROSE HEADER FILE (RH+INST NO)
41: SR.IT #HED1
42: AS 15=#HED1
43: AS 20=S1
44: $PR ENTER NAME OF CORFILE
45: $SR.IT #COR
46: AS 40=#COR
47: AS 6=T2
48: AS 30=1512ROSE*REVCAT.B
49: $PR ENTER FIRST DEMUX FILE
50: $SR.IT #DA
51: AS 61=#DA
52: $PR ENTER SECOND DEMUX FILE
53: $SR.IT #D2
54: AS 62=#D2
55: $PR ENTER THIRD DEMUX FILE
56: $SR.IT #D3
57: AS 63=#D3
58: $PR ENTER FOURTH DEMUX FILE
59: $SR.IT #D4
60: $AS 64=#D4
61: 1512ROSE*XBHIROS
62: WI 10
63: CO T2 10
64: FR 10
65: RW 11
66: SP REPORT :6
67: CO REPORT :6
68: PR Archive part 1 is finished. If you want to write this archive
69: PR to tape immediately, enter COPY:
70: SR.IT #QTN
71: IF (#QTN.NE."COPY") ME
72: JU !CMIN
73: !CMIN PR TAPE DRIVE 12
74: /PS 12
75: PR TAPE DRIVE 10
76: /PS 10
77: PR TAPE DRIVE 9
78: /PS 9
79: RW 11
80: PR ENTER DRIVE NO. TO RESOURCE
```

```
81:  $SR.IN #NUM
82:  SR.N #TAP=1
83:  !COPY $PR Writing to a blank tape?  Yes or no:
84:  $SR.IT #BLK
85:  IF,(#TAP>1) JU !NXT
86:  $PR Mount ARCHIVE TAPE with WRITE RING-Check screen for ARCTAP
87:  $RS 4=ARCTAP 1600B WR WA :#NUM
88:  !NXT $PR Enter number of outfiles to write to this tape:
89:  $SR.IN #NOF
90:  $SR.N #CNT=0
91:  IF,(#BLK="YES") $JU !RES
92:  1512ROSE*XTAPOS
93:  !RES $SR.N #CNT=#CNT+1
94:  $IF,(#CNT>#NOF) $JU !REW
95:  $PR ENTER NAME OF OUTFILE #CNT
96:  $SR.IT #FIL
97:  AS 20=#FIL
98:  $ $XCOMPAR
99:  PR If headers do not match, interrupt macro and check them
100: !WRIT $CO 11 4 BB TB=224 REC 1
101: $PR TAPE HEADER FOR #FIL NOW WRITTEN TO ARCTAP
102: WE 4
103: $PR NOW WRITING #FIL TO TAPE
104: CO #FIL 4 BB TB=4144 ALL
105: $PR PAU -  Write next Archive or Stop
106: $JU !RES
107: !REW $PR WRITE ANOTHER TAPE? YES OR NO.
108: SR.IT #ANS
109: #TAP=#TAP+1
110: IF,(#ANS="YES") JU !COPY
111: RW 4
112: PR TO DO A CHECK READ OF THE ARCHIVE TAPE, ASSIGN 10=SCREEN OR LISTOUT
113: PR FILE AND RUN XDISTP. OTHERWISE, FREE 4, ELIMINATE OUTFILES,
114: $PR DISMOUNT TAPE, LOG & STORE
115: $ME
```

```
100
 1:       NAME ROSEHD
 2: C PROGRAM 'ROSEHD' TO ENTER THE ROSE FORMAT TAPE HEADER FILE AND
 3: C CERTAIN FILE HEADER CHANNEL PARAMETERS.
 4: C                SLL 10/11/79
 5: C     LAST MODIFIED 1/29/81
 6: C
 7:       DIMENSION MBUF(256)
 8:       INTEGER IID(2),DNAME(28),EXCODE(10),STIME(5),ETIME(5),NFILES(4)
 9:       INTEGER CPAR(20,4)
10:       COMMON/HDR/ IID,DNAME,EXCODE,STIME,ETIME,NFILES,CPAR
11:       EQUIVALENCE (IID,MBUF)
12: C
13: C
14: C  BUILD TAPE HEADER FILE PLUS CHANNEL PARAMETERS
15: C
16:       WRITE(3,7990)
17: 7990  FORMAT(' HIG (1) OR NON-HIG (0)?:')
18:       READ(0,) KHIG
19:       WRITE(3,8000)
20: 8000  FORMAT(1X,'ENTER ROSE INSTRUMENT # (4 CHAR):')
21:       READ(0,8001) (IID(I),I=1,2)
22: 8001  FORMAT(2R2)
23:       WRITE(3,8002)
24: 8002  FORMAT(1X,'ENTER DESIGNER NAME & ADDRESS (56 CHAR):')
25:       READ(0,8003) (DNAME(I),I=1,28)
26: 8003  FORMAT(28R2)
27:       WRITE(3,8004)
28: 8004  FORMAT(1X,'ENTER EXPERIMENT CODE (20 CHAR):')
29:       READ(0,8005) (EXCODE(I),I=1,10)
30: 8005  FORMAT(10R2)
31:       WRITE(3,8006)
32: 8006  FORMAT(1X,'ENTER DATA START TIME(YR,MO,DA,HR,MN)-NO SPACES,',
33:      +'I.E., 7902210530')
34:       READ(0,8007) STIME
35: 8007  FORMAT(5R2)
36:       WRITE(3,8008)
37: 8008  FORMAT(1X,'ENTER DATA END TIME(YR,MO,DA,HR,MN)')
38:       READ(0,8007) ETIME
39:       WRITE(3,8009)
40: 8009  FORMAT(1X,'ENTER # FILES ON TAPE')
41:       READ(0,8010) NFILES
42: 8010  FORMAT(4R2)
43:       IF(KHIG.EQ.1)
44:       .  FOR I=1,4
45:       .  .  WRITE(3,8011) I
46: 8011  .  .  FORMAT(1X,'ENTER CODE FOR CHANNEL # ',I1,/'(1=V,',
47:      +.  .  '2=R,3=T,4=P,5=H1,6=H2,7=WW,8=TI)')
48:       .  .  READ(0,) CPAR(1,I)
49:       .  .  WRITE(3,8013)I
50: 8013  .  .  FORMAT(1X,'RCVR SENSITIVITY AT F0,MV/CM/SEC(CH ',I1,'):')
51:       .  .  READ(0,) CPAR(3,I)
52:       .  .  CPAR(2,I)=0
53:       .  .  WRITE(3,8014)I
54: 8014  .  .  FORMAT(1X,'ENTER F0,MILLIHZ(CH ',I1,'):')
55:       .  .  READ(0,) CPAR(4,I)
56:       .  .  WRITE(3,8012)I
57: 8012  .  .  FORMAT(1X,'RCVR AMPL FREQ CUTOFFS(HZ);LOW,HI(CH ',I1,'):')
58:       .  .  READ(0,) CPAR(5,I),CPAR(6,I)
59:       .  .  WRITE(3,8015)I
60: 8015  .  .  FORMAT(1X,'AMPL GAIN OF DIG FILTER IN DB(CH ',I1,'):')
61:       .  .  READ(0,) CPAR(7,I)
62:       .  .  WRITE(3,8016)I
63: 8016  .  .  FORMAT(1X,'DIG FILTER CUTOFF FREQS; LOW,HIGH(CH ',I1,'):')
64:       .  .  READ(0,) CPAR(8,I),CPAR(9,I)
65:       .  .  WRITE(3,8017)I
66: 8017  .  .  FORMAT(1X,'LOW ORDER BIT IN MICROVOLTS(CH ',I1,'):')
67:       .  .  READ(0,) CPAR(10,I)
68:       .  .  CPAR(11,I)=0
69:       .  .  CPAR(12,I)=0
70:       .  .  FOR J=13,20
71:       .  .  .  CPAR(J,I)=0
72:       .  .  END FOR
73:       .  END FOR
74:       .  FOR I=135,256
75:       .  .  MBUF(I)=0
76:       .  END FOR
77:       ELSE
78:       .  FOR I=55,256
79:       .  .  MBUF(I)=0
80:       .  END FOR
```

```
81:       END IF
82:       BUFFER OUT(15,MBUF,B,256,MSTAT,MLEN)
83:       CALL STATUS(15)
84:       ENDFILE 15
85:       STOP PAU
86:       END
```

```
102
 1: C PROGRAM DISCAT TO DISPLAY CONTENTS OF TAPE HEADER CATALOG AND
 2: C OPTIONALLY EVENT CATALOG DATA ON ALL EVENTS OF A GIVEN TAPE
 3: C
 4: C    WRITTEN BY SHARON LATRAILLE  HIG 363 X7796   LAST UPDATED 4/27/82
 5: C          TO COMPILE USE J.DISC
 6: C
 7:       NAME DISCAT
 8:       INTEGER CATBUF(112),THBUF(224),TRAY(7),RDATE(3),JDATE(3)
 9:       INTEGER IDATE(3),IDAT1(3),IDAT2(3),KDATE(3),COMBUF(24)
10:       INTEGER RERR,ELOER,BUB,SAMP,FNUM,INUM,ENUM,CRA,CCODE
11:       INTEGER WDEPI,WDEPE,IDEP,EDEP,ICHN,EXPL,TYPE
12:       INTEGER CATLFN,DEPIMN,DEPEMX,DEPEMN
13:       INTEGER ENUMMX,ENUMMN,EXPLMX,EXPLMN,DEPEMX,DEPEMN,ICHNMX
14:       INTEGER ICHNMN,TYPEMX,TYPEMN,EDEPMX,EDEPMN
15:       REAL ILAT,ILON,ILATMN,ILONMN
16:       DIMENSION PLACE(5),DNAME(9),EXCODE(4),SHTLN(10)
17:       INTEGER*6 SBT,CSEC,DST,DSTMAX,DSTMIN,TDST,TDET
18:       EQUIVALENCE (CATBUF(1),ITYPE),(CATBUF(2),JTAP),(CATBUF(3),INUM)
19:       EQUIVALENCE (CATBUF(4),ENUM),(CATBUF(7),SBT),(CATBUF(22),WDEPI)
20:       EQUIVALENCE (CATBUF(5),DST),(CATBUF(9),SIZE),(CATBUF(11),RANGE)
21:       EQUIVALENCE (CATBUF(13),ILAT),(CATBUF(15),ILON),(CATBUF(17),ELAT)
22:       EQUIVALENCE (CATBUF(19),ELON),(CATBUF(21),EXPL)
23:       EQUIVALENCE (CATBUF(23),WDEPE),(CATBUF(24),IDEP),(CATBUF(25),EDEP)
24:       EQUIVALENCE (CATBUF(28),RERR),(CATBUF(29),ELOER),(CATBUF(30),BUB)
25:       EQUIVALENCE (CATBUF(31),SAMP),(CATBUF(32),NWDS)
26:       EQUIVALENCE (CATBUF(33),FNUM),(CATBUF(34),NREC),(CATBUF(35),NSAM)
27:       EQUIVALENCE (CATBUF(36),IDEL),(CATBUF(37),IDAT1)
28:       EQUIVALENCE (CATBUF(27),TYPE),(CATBUF(67),SHTLIN)
29:       EQUIVALENCE (CATBUF(26),ICHN),(CATBUF(43),COMBUF)
30:       EQUIVALENCE (CATBUF(40),IDAT2),(CATBUF(69),CCODE)
31:       COMMON/ITCM/TRAY,CSEC,JULD
32:       DATA CATLFN/30/
33: C
34:       OPEN CATLFN
35:       WRITE(3,2000)
36: 2000  FORMAT(1X,'INPUT ROSE ARCHIVE TAPE # AND DASH #:')
37:       READ(0,)JTAPE,IDASH
38:       WRITE(3,1999)
39: 1999  FORMAT(1X,'DO YOU WANT DISPLAY OF EVENTS ON THIS TAPE?',
40:      +' YES=1:')
41:       READ(0,) KEY1
42:       WRITE(3,1998)
43: 1998  FORMAT(1X,'WANT TAPE HEADER BUFFERED OUT TO FILE? YES=1:')
44:       READ(0,) KEY2
45:       CALL DPOS(CATLFN,2)
46:       CALL BUFIN(CATLFN,CATBUF,112,IEOF)
47:       CRA=CATBUF(112)
48:         WRITE(3,) CRA
49:       CALL DPOS(CATLFN,CRA)
50:       LOOP
51: 1     .  CALL BUFIN(CATLFN,THBUF,224,IEOF)
52:       .  EXIT LOOP IF (IEOF.EQ.3)
53:       .  DECODE(9,4000,THBUF) ITAP
54:       .  DECODE(3,6003,THBUF(160)) IDSH
55:       .  INEXT=THBUF(153)
56: 4000  .  FORMAT(3X,I6)
57:       .  WRITE(10,) JTAPE,ITAP,INEXT
58:       .  IF(ITAP.EQ.JTAPE)
59:       .  .  IF(IDSH.NE.IDASH) GO TO 1
60:       .  .  IF(KEY2.EQ.1)
61:       .  .  .  BUFFER OUT(11,THBUF,B,224,MSTAT,MLEN)
62:       .  .  .  CALL STATUS(11)
63:       .  .  END IF
64:       .  .  DECODE(75,4001,THBUF(4)) NSLN,(PLACE(I),I=1,5),IDOC,
65:      +.  .  (RDATE(I),I=1,3),(IDATE(I),I=1,3),(JDATE(I),I=1,3),
66:      +.  .  (KDATE(I),I=1,3)
67: 4001  .  .  FORMAT(I6,5A6,13A3)
68:       .  .  DECODE(96,6000,THBUF(29)) INUMMN,ENUMMX,ENUMMN,DSTMAX,
69:      +.  .  DSTMIN,EXPLMX,EXPLMN,DEPIMN,DEPEMX,DEPEMN,
70:      +.  .  IDEPMN,EDEPMX,EDEPMN,ICHNMX,ICHNMN,TYPEMX,TYPEMN
71: 6000  .  .  FORMAT(3I6,2I13,2I2,6I6,2I3,2I2,2X)
72:       .  .  DECODE(102,6001,THBUF(61)) SIZEMX,SIZEMN,RANGMX,RANGMN,
73:      +.  .  ILATMN,ILONMN,ELATMX,ELATMN,ELONMX,ELONMN
74: 6001  .  .  FORMAT(10F10.4,2X)
75:       .  .  DECODE(174,6002,THBUF(95)) IID,DNAME,EXCODE,NFILES,TDST,TDET,SH
76: 6002  .  .  FORMAT(I4,12A6,A2,I6,I14,1X,I14,1X,10A6)
77:       .  .  DECODE(3,6003,THBUF(160)) IDASH
78: 6003  .  .  FORMAT(I3)
79:       .  .  INEXT=THBUF(153)
80:       .  .  WRITE(3,100) INEXT
```

```
 81: 100   . .  FORMAT(1X,'Data start address= ',I8)
 82:       . .  WRITE(10,2001) ITAP,IDASH
 83: 2001  . .  FORMAT(1X,'ROSE ARCHIVE TAPE NO.',T40,I6,' -',I2)
 84:       . .  IF(NSLN.EQ.1)
 85:       . .  .  WRITE(10,2002) SHTLN(1)
 86: 2002  . .  .  FORMAT(1X,'SHOT LINE NO.',T40,A6)
 87:       . .  END IF
 88:       . .  IF(NSLN.GT.1)
 89:       . .  .  WRITE(10,2500) SHTLN
 90: 2500  . .  .  FORMAT(1X,'SHOT LINE NOS: ',T40,A6,9(' ',A6))
 91:       . .  END IF
 92:       . .  WRITE(10,2003) PLACE
 93: 2003  . .  FORMAT(1X,'INSTITUTION RECD TAPE FROM',T40,5A6)
 94:       . .  WRITE(10,2004) IDOC
 95: 2004  . .  FORMAT(1X,'DOCUMENTATION CODE; 1=YES',T40,A3)
 96:       . .  WRITE(10,2005) RDATE
 97: 2005  . .  FORMAT(1X,'DATE ARCHIVE TAPE RECEIVED',T40,3A3)
 98:       . .  WRITE(10,2006) IDATE
 99: 2006  . .  FORMAT(1X,'DATE ARCHIVED',T40,3A3)
100:       . .  WRITE(10,2007) JDATE
101: 2007  . .  FORMAT(1X,'DATE LAST UPDATED',T40,3A3)
102:       . .  WRITE(10,2008) KDATE
103: 2008  . .  FORMAT(1X,'DATE LAST ACCESSED',T40,3A3)
104:       . .  WRITE(10,2009)
105: 2009  . .  FORMAT(1X,'************** MINIMUM & MAXIMUM VALUES OF ',
106:     +.  .  'KEYWORDS **************')
107:       . .  WRITE(10,2010) INUMMN
108: 2010  . .  FORMAT(1X,'INSTRUMENT NUMBER',T40,I6)
109:       . .  WRITE(10,2011) ENUMMN,ENUMMX
110: 2011  . .  FORMAT(1X,'EVENT NUMBERS',T40,2I6)
111:       . .  CSEC=DSTMIN
112:       . .  CALL CNTITM
113:       . .  WRITE(10,2012) TRAY
114: 2012  . .  FORMAT(1X,'MINIMUM DATA START TIME',T40,7I4)
115:       . .  CSEC=DSTMAX
116:       . .  CALL CNTITM
117:       . .  WRITE(10,2013) TRAY
118: 2013  . .  FORMAT(1X,'MAXIMUM DATA START TIME',T40,7I4)
119:       . .  WRITE(10,2014) EXPLMN,EXPLMX
120: 2014  . .  FORMAT(1X,'EXPLOSIVE TYPES',T40,2I2)
121:       . .  WRITE(10,2015) DEPTMN
122: 2015  . .  FORMAT(1X,'WATER DEPTH AT INSTRUMENT',T40,I6)
123:       . .  WRITE(10,2016) DEPEMN,DEPEMX
124: 2016  . .  FORMAT(1X,'WATER DEPTHS AT EVENT',T40,2I6)
125:       . .  WRITE(10,2017) IDEPMN
126: 2017  . .  FORMAT(1X,'INSTRUMENT DEPTH',T40,I6)
127:       . .  WRITE(10,2018) EDEPMN,EDEPMX
128: 2018  . .  FORMAT(1X,'EVENT DEPTHS',T40,2I6)
129:       . .  WRITE(10,2019) ICHNMN,ICHNMX
130: 2019  . .  FORMAT(1X,'#S OF CHANNELS',T40,2I3)
131:       . .  WRITE(10,2020) TYPEMN,TYPEMX
132: 2020  . .  FORMAT(1X,'EVENTS TYPES',T40,2I2)
133:       . .  WRITE(10,2021) SIZEMN,SIZEMX
134: 2021  . .  FORMAT(1X,'EVENT SIZES',T40,2F10.4)
135:       . .  WRITE(10,2022) RANGMN,RANGMX
136: 2022  . .  FORMAT(1X,'RANGES',T40,2F10.4)
137:       . .  WRITE(10,2023) ILATMN
138: 2023  . .  FORMAT(1X,'INSTRUMENT LATITUDE',T40,F10.4)
139:       . .  WRITE(10,2024) ILONMN
140: 2024  . .  FORMAT(1X,'INSTRUMENT LONGITUDE',T40,F10.4)
141:       . .  WRITE(10,2025) ELATMN,ELATMX
142: 2025  . .  FORMAT(1X,'EVENT LATITUDE RANGE',T40,2F10.4)
143:       . .  WRITE(10,2026) ELONMN,ELONMX
144: 2026  . .  FORMAT(1X,'EVENT LONGITUDE RANGE',T40,2F10.4)
145:       . .  WRITE(10,2027)
146: 2027  . .  FORMAT(1X,'***** TAPE HEADER FILE CONTENTS *****')
147:       . .  WRITE(10,2028) IID,DNAME
148: 2028  . .  FORMAT(1X,'INSTR. #',T20,I4,'    DESIGNER',T40,9A6)
149:       . .  WRITE(10,2029) EXCODE,NFILES
150: 2029  . .  FORMAT(1X,'EXPERIMENT:',T15,3A6,A2,' # OF EVENTS:',
151:     +.  .  T50,I6)
152:       . .  CSEC=TDST
153:       . .  CALL CNTITM
154:       . .  WRITE(10,2030) TRAY
155: 2030  . .  FORMAT(1X,'TAPE DATA START TIME',T40,7I4)
156:       . .  CSEC=TDET
157:       . .  CALL CNTITM
158:       . .  WRITE(10,2031) TRAY
159: 2031  . .  FORMAT(1X,'TAPE DATA END TIME',T40,7I4)
160:       . .  EXIT LOOP
```

```
161:        .  END IF
162:        END LOOP
163: C
164: C DISPLAY EVENT CATALOG CONTENTS IF REQUESTED
165: C
166:        JJ=0
167:        IF(KEY1.EQ.1)
168:        .  CALL DPOS(CATLFN,INEXT)
169:        .  LOOP
170:        .  .  CALL BUFIN(CATLFN,CATBUF,112,IEOF)
171:        .  .  EXIT LOOP IF (IEOF.EQ.3)
172: C      .  DECODE(9,4999,CATBUF) ITYPE,JTAP
173: C4999  .  FORMAT(I3,I6)
174:        .  .  EXIT LOOP IF (ITYPE.EQ.2)
175:        .  .  IF(JTAP.EQ.JTAPE)
176:        .  .  .  JJ=JJ+1
177:        .  .  .  EXIT LOOP IF (JJ.GT.NFILES)
178:        .  .  .  WRITE(10,1000)
179: 1000   .  .  .  FORMAT(1X,'********  EVENT CATALOG LIST  **********')
180:        .  .  .  WRITE(10,1001) ENUM,INUM,SHTLIN
181: 1001   .  .  .  FORMAT(1X,'EVENT NO.',T20,I6,'     INSTRUMENT NO.',
182:       +.  .  .  T50,I6,T62,A6)
183:        .  .  .  CSEC=DST
184:        .  .  .  CALL CNTITM
185:        .  .  .  WRITE(10,1002) TRAY
186: 1002   .  .  .  FORMAT(1X,'DATA START TIME',T40,7I4)
187:        .  .  .  CSEC=SBT
188:        .  .  .  CALL CNTITM
189:        .  .  .  WRITE(10,1003) TRAY
190: 1003   .  .  .  FORMAT(1X,'EVENT INSTANT TIME',T40,7I4)
191:        .  .  .  IF(TYPE.EQ.1) GO TO 15
192:        .  .  .  WRITE(10,1004) SIZE
193: 1004   .  .  .  FORMAT(1X,'EVENT SIZE (KG)',T40,F10.4)
194:        .  .  .  GO TO 20
195: 15     .  .  .  WRITE(10,1005) SIZE
196: 1005   .  .  .  FORMAT(1X,'EVENT MAGNITUDE ',T40,F10.4)
197: 20     .  .  .  WRITE(10,1006) RANGE
198: 1006   .  .  .  FORMAT(1X,'EVENT TO INSTRUMENT RANGE (KM)',
199:       +.  .  .  T40,F10.4)
200:        .  .  .  WRITE(10,1007) ILAT,ILON
201: 1007   .  .  .  FORMAT(1X,'INSTRUMENT LAT,LON',T40,2F10.4)
202:        .  .  .  WRITE(10,1008) ELAT,ELON
203: 1008   .  .  .  FORMAT(1X,'EVENT LAT,LON',T40,2F10.4)
204:        .  .  .  IF(TYPE.EQ.2)
205:        .  .  .  .  WRITE(10,1009) EXPL
206: 1009   .  .  .  .  FORMAT(1X,'EXPLOSIVE TYPE',T40,I4)
207:        .  .  .  END IF
208:        .  .  .  WRITE(10,1010) WDEPI,IDEP
209: 1010   .  .  .  FORMAT(1X,'WATER DEPTH AT INSTRUMENT(MSEC),INST DEPTH(M)',
210:       +.  .  .  T50,2I6)
211:        .  .  .  WRITE(10,1011) WDEPE,EDEP
212: 1011   .  .  .  FORMAT(1X,'WATER DEPTH AT EVENT(M),EVENT DEPTH(M)',
213:       +.  .  .  T50,2I6)
214:        .  .  .  WRITE(10,1012) TYPE
215: 1012   .  .  .  FORMAT(1X,'EVENT TYPE; 1=EARTHQUAKE, 2=SHOT ',
216:       +.  .  .  T40,I4)
217:        .  .  .  WRITE(10,1013) RERR,ELOER
218: 1013   .  .  .  FORMAT(1X,'ERROR EST.IN RANGE(M)',T40,I6,
219:       +.  .  .  /' ERROR RADIUS IN EVENT LOCATION',T40,I6)
220:        .  .  .  IF(TYPE.EQ.2)
221:        .  .  .  .  WRITE(10,1014) BUB
222: 1014   .  .  .  .  FORMAT(1X,'BUBBLE PULSE PERIOD (MSEC)',T40,I6)
223:        .  .  .  END IF
224:        .  .  .  WRITE(10,1015) ICHN
225: 1015   .  .  .  FORMAT(1X,'NUMBER OF CHANNELS',T40,I4)
226:        .  .  .  WRITE(10,1023) (COMBUF(I),I=1,10)
227: 1023   .  .  .  FORMAT(1X,'CHANNEL CODES (0=NO CHANNEL)- CHANNELS 1 TO 10: '
228:       +.  .  .  T40,10I3)
229:        .  .  .  WRITE(10,1016) SAMP
230: 1016   .  .  .  FORMAT(1X,'SAMPLE RATE IN SAMP/SEC',T40,I6)
231:        .  .  .  WRITE(10,1017) NWDS
232: 1017   .  .  .  FORMAT(1X,'NUMBER OF WORDS/RECORD (SHOULD',
233:       +.  .  .  ' =4096',T40,I6)
234:        .  .  .  WRITE(10,1018) NREC
235: 1018   .  .  .  FORMAT(1X,'NUMBER OF RECORDS/COMPONENT',T40,I6)
236:        .  .  .  WRITE(10,1044) NSAM
237: 1044   .  .  .  FORMAT(1X,'NUMBER OF SAMPLES IN LAST RECORD',T40,I6)
238:        .  .  .  WRITE(10,1019) FNUM
239: 1019   .  .  .  FORMAT(1X,'FILE # WITHIN TAPE',T40,I6)
240:        .  .  .  WRITE(10,1020) IDEL
```

```
241: 1020  . . .  FORMAT(1X,'DELETE KEY (1=DELETED)',T40,I1)
242:       . . .  WRITE(10,1021) IDAT1
243: 1021  . . .  FORMAT(1X,'DATE LAST UPDATED',T40,3A3)
244:       . . .  WRITE(10,1022) IDAT2
245: 1022  . . .  FORMAT(1X,'DATE LAST ACCESSED',T40,3A3)
246:       . .  END IF
247:       .  END LOOP
248:       END IF
249:       STOP PAU
250:       END
```

```
C PROGRAM LISHDR LISTS ALL THE CATALOG FILE TAPE HEADERS
C
C          LAST MODIFIED 4/27/82  BY SLL
C                  AS 30=CATALOG
C                  AS 10=LISTOUT
C
      NAME LISHDR
      INTEGER CATBUF(112),THBUF(224),TRAY1(7),TRAY(7),RDATE(3),JDATE(3)
      INTEGER IDATE(3),KDATE(3)
      INTEGER CRA
      INTEGER CATLFN,DEPIMN,DEPEMX,DEPEMN
      INTEGER ENUMMX,ENUMMN,EXPLMX,EXPLMN,DEPEMX,DEPEMN,ICHNMX
      INTEGER ICHNMN,TYPEMX,TYPEMN,EDEPMX,EDEPMN
      REAL ILATMN,ILONMN
      DIMENSION PLACE(5),DNAME(9),EXCODE(4),SHTLN(10)
      INTEGER*6 CSEC,DSTMAX,DSTMIN,TDST,TDET
      COMMON/ITCM/TRAY,CSEC,JULD
      DATA CATLFN/30/
C
      OPEN CATLFN
      CALL DPOS(CATLFN,2)
      CALL BUFIN(CATLFN,CATBUF,112,IEOF)
      CRA=CATBUF(112)
      WRITE(3,1010)
1010  FORMAT(' LIST BY TAPE (1) OR BY INSTRUMENT (2):')
      READ(3,) KBY
      CALL DPOS(CATLFN,CRA)
      ICNT=10
      LOOP
      .  IF(ICNT.EQ.10)
      .  .  ICNT=0
      .  .  IF(KBY.EQ.2)
      .  .  .   WRITE(10,3001)
3001  .  .  .  FORMAT(/' INSTRUM.   LATITUDE   LONGITUDE    #FILES',
     +.  .  .  '          EVENTS         #SHOTLINES')
      .  .  ELSE
      .  .  .  WRITE(10,3000)
3000  .  .  .  FORMAT(/'   TAPE   IID   TAPE DST      TAPE DET   #FILES',
     +.  .  .  'EVENTS   ARCH DATE')
      .  .  END IF
      .  END IF
      .  CALL BUFIN(CATLFN,THBUF,224,IEOF)
      .  ICNT=ICNT+1
      .  EXIT LOOP IF (IEOF.EQ.3)
      .   DECODE(9,4000,THBUF) ITAP
4000  .   FORMAT(3X,I6)
      .   DECODE(3,4007,THBUF(160)) IDASH
4007  .  FORMAT(I3)
      .  DECODE(75,4001,THBUF(4)) NSLN,(PLACE(I),I=1,5),IDOC,
     +.  (RDATE(I),I=1,3),(IDATE(I),I=1,3),(JDATE(I),I=1,3),
     +.  (KDATE(I),I=1,3)
4001  .  FORMAT(I6,5A6,13A3)
      .  DECODE(96,6000,THBUF(29)) INUMMN,ENUMMX,ENUMMN,DSTMAX,
     +.  DSTMIN,EXPLMX,EXPLMN,DEPIMN,DEPEMX,DEPEMN,
     +.  IDEPMN,EDEPMX,EDEPMN,ICHNMX,ICHNMN,TYPEMX,TYPEMN
6000  .  FORMAT(3I6,2I3,2I2,6I6,2I3,2I2,2X)
      .  DECODE(102,6001,THBUF(61)) SIZEMX,SIZEMN,RANGMX,RANGMN,
     +.  ILATMN,ILONMN,ELATMX,ELATMN,ELONMX,ELONMN
6001  .  FORMAT(10F10.4,2X)
      .  DECODE(174,6002,THBUF(95)) IID,DNAME,EXCODE,NFILES,TDST,TDET,SHTLN
6002  .  FORMAT(I4,12A6,A2,I6,I14,1X,I14,1X,10A6)
      .  IF(KBY.EQ.1)
C SKIP TIME CODE FOR KBY=2
      .  .  CSEC=TDST
      .  .  CALL CNTITM
      .  .  FOR J=1,7
      .  .  .  TRAY1(J)=TRAY(J)
      .  .  END FOR
      .  .  TRAY1(1)=TRAY1(1)-1900
      .  .  JULD1=JULD
      .  .   CSEC=TDET
      .  .  CALL CNTITM
      .  .  TRAY(1)=TRAY(1)-1900
      .  .  WRITE(10,2501) ITAP,IDASH,IID,TRAY1(1),JULD1,TRAY1(4),TRAY1(5),
     +.  .  TRAY(1),JULD,TRAY(4),TRAY(5),NFILES,ENUMMN,ENUMMX,IDATE
2501  .  .  FORMAT(I3,' -',I2,I5,4I3,2X,4I3,2X,I5,I6,' -',I6,2X,3A3)
      .  ELSE
      .  .  WRITE(10,2502) IID,ILATMN,ILONMN,NFILES,ENUMMN,ENUMMX,NSLN
2502  .  .  FORMAT(I8,2F12.4,I10,2X,I6,' - ',I6,I10)
      .  END IF
```

```
81:       END LOOP
82:       STOP PAU
83:       END
```

```
 1: C PROGRAM RDHDR  TO READ THE CATALOG EVENT HEADERS
 2: C
 3: C*************************************************************************
 4: C   LAST UPDATED 6/16/82
 5: C
 6: C
 7:       NAME RDHDR
 8:       INTEGER COMBUF(24),CATBUF(112),JBUF(112),INUM,ENUM
 9:       INTEGER CATLFN,TAPENM,ICSEL(8),ISTAT(2),FNUM
10:       INTEGER RERR,ELOER,BUB,SAMP,TRAY(7),TYPE,CCODE
11:       INTEGER WDEPI,WDEPE,IDEP,EDEP,ICHN,HDLFN,EXPL
12:       REAL ILAT,ILON
13:       COMMON /ITCM/TRAY,CSEC,JULD
14:       EQUIVALENCE (JBUF(1),CATBUF(1))
15:       EQUIVALENCE (CATBUF(1),JTYPE),(CATBUF(2),TAPENM),(CATBUF(3),INUM)
16:       EQUIVALENCE (CATBUF(4),ENUM),(CATBUF(5),DST),(CATBUF(7),SBT)
17:       EQUIVALENCE (CATBUF(9),SIZE),(CATBUF(11),RANGE)
18:       EQUIVALENCE (CATBUF(13),ILAT),(CATBUF(15),ILON),(CATBUF(17),ELAT)
19:       EQUIVALENCE (CATBUF(19),ELON),(CATBUF(21),EXPL),(CATBUF(22),WDEPI)
20:       EQUIVALENCE (CATBUF(23),WDEPE),(CATBUF(24),IDEP),(CATBUF(25),EDEP)
21:       EQUIVALENCE (CATBUF(26),ICHN),(CATBUF(27),TYPE)
22:       EQUIVALENCE (CATBUF(28),RERR),(CATBUF(29),ELOER),(CATBUF(30),BUB)
23:       EQUIVALENCE (CATBUF(31),SAMP),(CATBUF(32),NWDS)
24:       EQUIVALENCE (CATBUF(33),FNUM),(CATBUF(34),NRC),(CATBUF(35),NSAM)
25:       EQUIVALENCE (CATBUF(36),IDEL),(CATBUF(37),IDATE)
26:       EQUIVALENCE (CATBUF(40),IDAT1),(CATBUF(43),COMBUF)
27:       EQUIVALENCE (CATBUF(67),SHTLIN),(CATBUF(69),CCODE)
28:       DATA LFNS/10/,HDLFN/30/
29: C
30:       CALL BTIME
31:       OPEN HDLFN
32:       WRITE(3,100)
33: 100   FORMAT(' ENTER START ADDRESS AND NO. OF EVENTS TO DISPLAY:')
34:       READ(0,) IAD,NREC
35:       IF(IAD.EQ.0) IAD=4
36:       IDT=0
37: 1     CALL DPOS(HDLFN,IAD)
38:       WRITE(LFNS,1000)
39: 1000  FORMAT(1X,' RECORD  TAPE# INST# EVENT  FNUM')
40:       LOOP(NREC)
41:       .  CALL BUFIN(HDLFN,CATBUF,112,IEOF)
42:       .  EXIT LOOP IF (IEOF.EQ.3)
43:       .  IF(IDT.EQ.1)
44:       .  .  CSEC=DST
45:       .  .  CALL CNTITM
46:       .  .  WRITE(LFNS,1111) TAPENM,INUM,ENUM,FNUM,(TRAY(I),I=2,5)
47: 1111  .  .  FORMAT(4I6,I6,'/',I2,I4,":",I2)
48:       .  .  GO TO 3
49:       .  END IF
50:       .  WRITE(LFNS,1100) TAPENM,INUM,ENUM,FNUM
51: 1100  .  FORMAT(4I6)
52: 3     END LOOP
53:       WRITE(3,101)
54: 101   FORMAT(' GO AGAIN? IF NO ENTER -1, IF SO ENTER START ADDRESS:')
55:       READ(0,)IAD
56:       WRITE(3,102)
57: 102   FORMAT(' DISPLAY TIME? Y=1,N=0:')
58:       READ(0,) IDT
59:       IF(IAD.LT.0) GO TO 99
60:       GO TO 1
61:  99   CALL ETIME
62:       ENDFILE LFNS
63:       STOP
64:       END
```

```
 1:       NAME FIX
 2: C PROGRAM TO FIX THE ROSE ARCHVIE CATALOG EVENT HEADER RECORDS
 3: C WRITTEN 7/8/80 S. LATRAILLE      LAST MODIFIED 12/6/82
 4: C
 5:       INTEGER GETLIN,CLINE(72),GETWRD,EOF,ARG(4),CTOI,FNDFLG
 6:       INTEGER CRA,CRAS,CATBUF(112),THBUF(224),CATLFN
 7:       INTEGER*6 ISTS
 8:       COMMON/BUFER/ THBUF,CATLFN,LFN
 9:       DATA CATLFN/30/,LFN/20/,EOF/10003/
10: C
11:       OPEN CATLFN
12:       OPEN LFN
13:       WRITE(3,5)
14: 5     FORMAT(' Program XFIX for fixing the Rose archive catalog',
15:      +' file event headers ',/' Enter (1) to change a small',
16:      +' number of events;',/' Enter (2) to apply same fix to',
17:      +' all events:',/' ENTER 3 TO READ FROM EXTERNAL FILE')
18:       READ(0,) ITYP
19:       WRITE(3,10)
20: 10    FORMAT(//' Enter Rose Archive Tape Numbers To Edit:(1 2 3 . . .)')
21:       NIL=GETLIN(CLINE,0)
22:       CALL DPOS(CATLFN,2)
23:       CALL BUFIN(CATLFN,CATBUF,112,IEOF)
24:       CRAS=CATBUF(112)
25:       WRITE(3,15)
26: 15    FORMAT(' Enter single dash #, or 0 for all dash #s:')
27:       READ(0,) KDSH
28: C   WRITE TITLES TO REPORT FILE
29:       WRITE(LFN,6001)
30: 6001  FORMAT(///'                CAT EVENT HEADER FIX LIST')
31:       ITEM=1
32:       WHILE(GETWRD(CLINE,ITEM,ARG).NE.EOF)
33:       . K=1
34:       . CALL DPOS(CATLFN,CRAS)
35:       . JTAPE=CTOI(ARG,K)
36:       . FNDFLG=0
37:       . INDEX=0
38: C LOCATE HEADER
39:       . LOOP
40:       . . CALL BUFIN(CATLFN,THBUF,224,IEOF)
41:       . . CALL DSTAT(CATLFN,ISTS,CRA)
42:       . . EXIT LOOP IF(IEOF.GE.3)
43:       . . DECODE(9,1001,THBUF) ITAP
44: 1001  . . FORMAT(3X,I6)
45:       . . IF(ITAP.EQ.JTAPE)
46:       . . . DECODE(3,1002,THBUF(160)) INDEX
47: 1002  . . . FORMAT(I3)
48: C      INDEX=INDEX+1
49:       . . . WRITE(3,20) ITAP,INDEX
50:       . . . IF (KDSH.NE.0)
51:       . . . . IF(INDEX.NE.KDSH) GO TO 88
52:       . . . END IF
53:       . . . WRITE(LFN,20) ITAP,INDEX
54: 20    . . . FORMAT(/' Tape #',I6,'-',I1)
55:       . . . FNDFLG=1
56:       . . . IF(ITYP.EQ.1)
57:       . . . . CALL CFIX4(CRA)
58:       . . . END IF
59:       . . . IF(ITYP.EQ.2)
60:       . . . . CALL CFIX5(CRA)
61:       . . . END IF
62:       . . . IF(ITYP.EQ.3)
63:       . . . . CALL CFIX8(CRA)
64:       . . . END IF
65:       . . END IF
66: 88    . END LOOP
67:       . IF(IEOF.GE.3.AND.FNDFLG.EQ.0)
68:       . . WRITE(3,40) JTAPE
69:       . . WRITE(LFN,40) JTAPE
70: 40    . . FORMAT(' ***ERROR: TAPE NUMBER',I6,'  NOT FOUND')
71:       . END IF
72:       . ITEM=ITEM+1
73:       END WHILE
74:       STOP
75:       END

 1:       SUBROUTINE CFIX4(CRA)
 2: C LAST MODIFIED 11/30/82
 3:       COMMON /BUFER/ THBUF,CATLFN,LFN
 4:       COMMON /ITCM/ TRAY,CSEC,JULD
```

```
110
 5:        REAL ARAY(6)
 6:        INTEGER CATBUF(112),CATLFN,THBUF(224),ENUM,ISTAT(2)
 7:        INTEGER TRAY(7),IDATE(3),IWD(10),EF,FNUM,CRA
 8:        INTEGER*6 CSEC,CS,CS1
 9:        EQUIVALENCE (CATBUF(1),ITYPE),(CATBUF(2),JTAP),(CATBUF(3),INUM)
10:        EQUIVALENCE (CATBUF(4),ENUM),(CATBUF(5),CS),(CATBUF(7),CS1)
11:        EQUIVALENCE (CATBUF(9),ARAY(1)),(CATBUF(33),FNUM)
12:        EQUIVALENCE (CATBUF(26),ICHN),(CATBUF(67),SHTLN)
13:        CALL DATE(IDATE)
14:        DECODE(6,4001,THBUF(121)) NFILES
15: 4001   FORMAT(I6)
16:        INEXT=THBUF(153)
17:        WRITE(3,95)
18: 95     FORMAT(' FILE #S (1) OR EVENT #S (2)?')
19:        READ(0,) KKEY
20:        WRITE(3,100)
21: 100    FORMAT(' ENTER BEG AND END FILE OR EVENT NUMBERS:')
22:        READ(0,) IBEG,IEND
23:        WRITE(3,101)
24: 101    FORMAT(' ENTER # OF WORDS TO BE CHANGED:',
25:       +/'    MAXIMUM IS 10')
26:        READ(0,) N
27:        WRITE(3,102)
28: 102    FORMAT(' ENTER WORD #S (W1 W2 W3 . . .):',
29:       +/' WORDS 37-42 NOT ALLOWED. See file EVENT for word list.')
30:        READ(0,) (IWD(I),I=1,N)
31: C POSITION TO EVENT HEADERS
32:        CALL DPOS(CATLFN,INEXT)
33:        ICRA=INEXT
34:        LOOP(NFILES)
35:        .  CALL BUFIN(CATLFN,CATBUF,112,IEOF)
36:        .  LRA=ICRA
37:        .  IF(KKEY.EQ.2)
38:        .  .  EF=ENUM
39:        .  ELSE
40:        .  .  EF=FNUM
41:        .  END IF
42:        .  IF(EF.GE.IBEG.AND.EF.LE.IEND)
43:        .  .  FOR I=1,N
44:        .  .  .  K=IWD(I)
45:        .  .  .  IF(K.GT.4.AND.K.LT.21.OR.K.GT.42.AND.K.LT.69)
46: C SPECIAL HANDLING
47:        .  .  .  .  IF(K.EQ.5)
48:        .  .  .  .  .  CSEC=CS
49:        .  .  .  .  .  CALL CNTITM
50:        .  .  .  .  .  WRITE(3,) TRAY
51:        .  .  .  .  .  READ(0,) TRAY
52:        .  .  .  .  .  CALL ITMCNT
53:        .  .  .  .  .  CS=CSEC
54:        .  .  .  .  END IF
55:        .  .  .  .  IF(K.EQ.7)
56:        .  .  .  .  .  CSEC=CS1
57:        .  .  .  .  .  CALL CNTITM
58:        .  .  .  .  .  WRITE(3,) TRAY
59:        .  .  .  .  .  READ(0,) TRAY
60:        .  .  .  .  .  CALL ITMCNT
61:        .  .  .  .  .  CS1=CSEC
62:        .  .  .  .  END IF
63: C      IF(K.EQ.9)
64: C      J=1
65: C      WRITE(3,107)FNUM,ENUM,K,ARAY(J)
66: C      LOOP
67: C      READ(40,125) ISN,SIZ
68: C25    FORMAT(I4,41X,F10.3)
69: C      EXIT LOOP IF(ISN.EQ.ENUM)
70: C      END LOOP
71: C      ARAY(J)=SIZ
72: C      WRITE(3,125) ISN,SIZ
73: C      END IF
74:        .  .  .  .  IF(K.GE.9.AND.K.LE.19)
75:        .  .  .  .  .  J=(K-9)/2+1
76:        .  .  .  .  .  WRITE(3,107)FNUM,ENUM,K,ARAY(J)
77: 107    .  .  .  .  .  FORMAT(' FILE',I4,' EVENT',I5,' WORD',I3,2X,F10.4,
78:       +.  .  .  .  .  'NEW VALUE')
79:        .  .  .  .  .  READ(0,) ARAY(J)
80:        .  .  .  .  END IF
81:        .  .  .  .  IF(K.EQ.43)
82:        .  .  .  .  .  WRITE(3,110) ENUM
83: 110    .  .  .  .  .  FORMAT(' EVENT ',I5)
84:        .  .  .  .  .  FOR J=1,ICHN
85:        .  .  .  .  .  .  WRITE(3,109) J,CATBUF(J+42)
```

```
 86: 109   . . . . . . FORMAT(' CHAN',I3,2X,I3,'  ENTER CODE:')
 87:       . . . . . . READ(0,) CATBUF(J+42)
 88:       . . . . . END FOR
 89:       . . . . END IF
 90:       . . . . IF(K.EQ.67)
 91:       . . . . . WRITE(3,104) ENUM,SHTLN
 92: 104   . . . . . FORMAT(' EVENT,SHTLN:',I6,1X,A6,' ENTER SHOTLINE N
 93:       . . . . . READ(0,8002) SHTLN
 94: 8002  . . . . . FORMAT(A6)
 95:       . . . . END IF
 96:       . . . ELSE
 97: C REGULAR HANDLING
 98:       . . . . WRITE(3,105) FNUM,ENUM,K,CATBUF(K)
 99: 105   . . . . FORMAT(' F#',I3,'EVENT',I5,'  WORD(',I2,') =',I7,'  E
100:      +. . . . 'NEW VALUE:')
101:       . . . . READ(0,) CATBUF(K)
102:       . . . END IF
103:       . . END FOR
104:       . . CALL DPOS(CATLFN,LRA)
105:       . . FOR L=1,3
106:       . . . CATBUF(L+36)=IDATE(L)
107:       . . . CATBUF(L+39)=IDATE(L)
108:       . . END FOR
109:       . . CALL BUFOUT(CATLFN,CATBUF,112,IEOF)
110:       . END IF
111:       . CALL DSTAT(CATLFN,ISTAT,ICRA)
112:       END LOOP
113: 99    CALL DPOS(CATLFN,CRA)
114:       RETURN
115:       END

  1:       SUBROUTINE CFIX5(CRA)
  2: C THIS SUBROUTINE ALLOWS MAKING WORD CHANGES TO ALL EVENTS
  3: C
  4: C LAST MODIFIED 4/20/82
  5:       COMMON /BUFER/ THBUF,CATLFN,LFN
  6:       COMMON /ITCM/ TRAY,CSEC,JULD
  7:       REAL ARAY(6),REAL(6)
  8:       INTEGER CATBUF(112),CATLFN,THBUF(224),ENUM,ISTAT(2)
  9:       INTEGER TRAY(7),IDATE(3),IWD(10),IC(24),CRA
 10:       INTEGER*6 CSEC,CS,CS1
 11:       EQUIVALENCE (CATBUF(1),ITYPE),(CATBUF(2),JTAP),(CATBUF(3),INUM)
 12:       EQUIVALENCE (CATBUF(4),ENUM),(CATBUF(5),CS),(CATBUF(7),CS1)
 13:       EQUIVALENCE (CATBUF(9),ARAY(1))
 14:       EQUIVALENCE (CATBUF(26),ICHN),(CATBUF(67),SHTLN)
 15:       CALL DATE(IDATE)
 16:       DECODE(6,4001,THBUF(121)) NFILES
 17: 4001  FORMAT(I6)
 18:       INEXT=THBUF(153)
 19:       LCNT=0
 20:       WRITE(3,101)
 21: 101   FORMAT(' ENTER # OF WORDS TO BE CHANGED:',
 22:      +/'    MAXIMUM IS 10')
 23:       READ(0,) N
 24:       WRITE(3,102)
 25: 102   FORMAT(' ENTER WORD #S (W1 W2 W3 . . .):',
 26:      +/' ** See file EVENT for word list.',
 27:      +/' NOTE: The following words are not allowed - ',
 28:      +/' Word #s 4,5,7,11,17,19,23,25,30,33,37,39.')
 29:       READ(0,) (IWD(I),I=1,N)
 30:       WRITE(3,100) IWD
 31: 100   FORMAT(' WORDS ENTERED:',10I4)
 32: C     CALL CHECK(N,IWD)
 33: C POSITION TO EVENT HEADERS
 34:       CALL DPOS(CATLFN,INEXT)
 35:       ICRA=INEXT
 36:       LOOP(NFILES)
 37:       . CALL BUFIN(CATLFN,CATBUF,112,IEOF)
 38:       . LRA=ICRA
 39:       . FOR I=1,N
 40:       . . K=IWD(I)
 41:       . . IF(K.GT.4.AND.K.LT.21.OR.K.GT.42.AND.K.LT.69)
 42: C SPECIAL HANDLING
 43:       . . . IF(K.GE.9.AND.K.LE.15)
 44:       . . . . J=(K-9)/2+1
 45:       . . . . IF(LCNT.EQ.0)
 46:       . . . . . WRITE(3,107)ENUM,K,ARAY(J)
 47: 107   . . . . . FORMAT(' EVENT',I5,' WORD',I3,2X,F10.4,'  ENTER NE
 48:       . . . . . READ(0,) ARAY(J)
 49:       . . . . . REAL(I)=ARAY(J)
```

```
50:       . . . . ELSE
51:       . . . . . ARAY(J)=REAL(I)
52:       . . . . END IF
53:       . . . END IF
54:       . . . IF(K.EQ.43)
55:       . . . . WRITE(3,110) ENUM
56: 110   . . . . FORMAT(' EVENT ',I5)
57:       . . . . FOR J=1,ICHN
58:       . . . . . IF(LCNT.EQ.0)
59:       . . . . . . WRITE(3,109) J,CATBUF(J+42)
60: 109   . . . . . . FORMAT(' CHAN',I3,2X,I3,'  ENTER CODE:')
61:       . . . . . . READ(0,) CATBUF(J+42)
62:       . . . . . . IC(J)=CATBUF(J+42)
63:       . . . . . ELSE
64:       . . . . . . CATBUF(J+42)=IC(J)
65:       . . . . . END IF
66:       . . . . END FOR
67:       . . . END IF
68:       . . . IF(K.EQ.67)
69:       . . . . IF(LCNT.EQ.0)
70:       . . . . . WRITE(3,104) ENUM,SHTLN
71: 104   . . . . . FORMAT(' EVENT,SHTLN:',I6,1X,A6,' ENTER SHOTLINE N
72:       . . . . . READ(0,8002) SHTLN
73: 8002  . . . . . FORMAT(A6)
74:       . . . . . SLINE=SHTLN
75:       . . . . ELSE
76:       . . . . . SHTLN=SLINE
77:       . . . . END IF
78:       . . . END IF
79:       . . ELSE
80: C REGULAR HANDLING
81:       . . . IF(LCNT.EQ.0)
82:       . . . . WRITE(3,105) ENUM,K,CATBUF(K)
83: 105   . . . . FORMAT(' EVENT',I5,'  WORD(',I2,') =',I7,'  ENTER NEW
84:       . . . . READ(0,) CATBUF(K)
85:       . . . . THBUF(I)=CATBUF(K)
86:       . . . ELSE
87:       . . . . CATBUF(K)=THBUF(I)
88:       . . . END IF
89:       . . END IF
90:       . END FOR
91:       . LCNT=1
92:       . CALL DSTAT(CATLFN,ISTAT,ICRA)
93:       . CALL DPOS(CATLFN,LRA)
94:       . FOR L=1,3
95:       . . CATBUF(L+36)=IDATE(L)
96:       . . CATBUF(L+39)=IDATE(L)
97:       . END FOR
98:       . CALL BUFOUT(CATLFN,CATBUF,112,IEOF)
99:       END LOOP
100: 99   CALL DPOS(CATLFN,CRA)
101:      RETURN
102:      END

  1:      SUBROUTINE CFIX8(CRA)
  2: C LAST MODIFIED 9/25/82   TO READ TIME & EVENT SIZE
  3: C AND OTHER PARAMETERS FROM SORTSLN FILE - ASSIGN 40 TO IT
  4: C CS=DATA START TIME, CS1=EVENT TIME
  5:      COMMON /BUFER/ THBUF,CATLFN,LFN
  6:      COMMON /ITCM/ TRAY,CSEC,JULD
  7:      INTEGER CATBUF(112),CATLFN,THBUF(224),ENUM,ISTAT(2)
  8:      INTEGER TRAY(7),IDATE(3),EDEP,WDEPE,EXP,CRA
  9:      INTEGER*6 CSEC,CS,CS1
 10:      EQUIVALENCE (CATBUF(1),ITYPE),(CATBUF(2),JTAP),(CATBUF(3),INUM)
 11:      EQUIVALENCE (CATBUF(4),ENUM),(CATBUF(5),CS),(CATBUF(7),CS1)
 12:      EQUIVALENCE (CATBUF(9),SIZE),(CATBUF(11),RANGE),(CATBUF(25),EDEP)
 13:      EQUIVALENCE (CATBUF(17),ELAT),(CATBUF(19),ELON),(CATBUF(21),EXP)
 14:      EQUIVALENCE (CATBUF(23),WDEPE),(CATBUF(30),IBUB)
 15:      EQUIVALENCE (CATBUF(26),ICHN),(CATBUF(67),SHTLN)
 16:        REWIND 40
 17:      CALL DATE(IDATE)
 18:      DECODE(6,4001,THBUF(121)) NFILES
 19: 4001 FORMAT(I6)
 20:      INEXT=THBUF(153)
 21:      WRITE(3,100)
 22: 100  FORMAT(' ENTER BEG AND END EVENT NUMBER:')
 23:      READ(0,) IBEG,IEND
 24: C POSITION TO EVENT HEADERS
 25:      CALL DPOS(CATLFN,INEXT)
 26:      ICRA=INEXT
```

```
27:       LOOP(NFILES)
28:       .  CALL BUFIN(CATLFN,CATBUF,112,IEOF)
29:       .  LRA=ICRA
30:       .  IF(ENUM.GE.IBEG.AND.ENUM.LE.IEND)
31: C SPECIAL HANDLING - EVENT TIME AND DST, AND SIZE
32: 1     .  .  READ(40,777) ISN,(TRAY(KK),KK=1,5),SEC,ELAT,ELON,SIZE,REDEP
33:      +.  .  IWDEMS,IBUB,EXP
34: 777   .  .  FORMAT(I4,5I3,F7.3,F9.4,F10.4,F10.3,F6.1,I6,I5,I2)
35:       .  .  EDEP=INT(REDEP)
36:       .  .  WDEPE=INT(IWDEMS*1.5)
37:       .  .  IF(ISN.NE.ENUM) GO TO 1
38: C     SEC =SEC+1.95
39: C     TRAY(6)=IFIX(SEC)
40: C     TRAY(7)=IFIX(SEC*1000.)-(TRAY(6)*1000)
41: C     CALL ITMCNT
42: C     CS1=CSEC
43: C     CS=CS1-10000
44:       .  .  GO TO 15
45: C REGULAR HANDLING
46: 15    .  .  CALL DPOS(CATLFN,LRA)
47:       .  .  FOR L=1,3
48:       .  .  .  CATBUF(L+36)=IDATE(L)
49:       .  .  .  CATBUF(L+39)=IDATE(L)
50:       .  .  END FOR
51:       .  .  CALL BUFOUT(CATLFN,CATBUF,112,IEOF)
52:       .  END IF
53:       .  CALL DSTAT(CATLFN,ISTAT,ICRA)
54:       END LOOP
55: 99    CALL DPOS(CATLFN,CRA)
56:       RETURN
57:       END
```

```
      NAME ROSED
C PROGRAM TO EDIT THE ROSE ARCHIVE CATALOG TAPE HEADER RECORDS
C WRITTEN 7/1/80 S. LATRAILLE          LAST UPDATED 5/13/82
C
C*************************************************************************
C                 Assignments: 30=Catalog file
C                              20=List out file
C
      INTEGER TAPNO(100),EOF,ARG(4),CTOI,FNDFLG
      INTEGER CRA,CRAS,CATBUF(112),THBUF(224),CATLFN
      INTEGER*6 ISTS
      COMMON/BUFER/ THBUF,CATLFN,LFN
      DATA CATLFN/30/,LFN/20/,EOF/10003/
C
      OPEN CATLFN
      OPEN LFN
      WRITE(3,5)
5     FORMAT(' Program XROSED for editing the Rose archive catalog',
     +' file tape header records; Rev 4, 5/13/82')
      WRITE(3,100)
100   FORMAT(' Enter # of tapes to edit:')
      READ(0,) KTN
      WRITE(3,10)
10    FORMAT(//' Enter Rose Archive Tape Numbers To Edit:(1 2 3 . . .)')
      READ(0,) (TAPNO(J),J=1,KTN)
      WRITE(3,11)
11    FORMAT(/' Enter dash-number, or 0 if all:')
      READ(0,) IDSH
      CALL DPOS(CATLFN,2)
      CALL BUFIN(CATLFN,CATBUF,112,IEOF)
      CRAS=CATBUF(112)
C   WRITE TITLES TO REPORT FILE
      WRITE(LFN,6001)
6001  FORMAT(///'                CATALOG FILE EDIT REPORT')
      ITEM=1
      K=1
      WHILE (KTN.GT.0)
      .  CALL DPOS(CATLFN,CRAS)
      .  JTAPE=TAPNO(K)
      .  FNDFLG=0
      .  INDEX=0
C LOCATE HEADER
      .  LOOP
      .  .  CALL BUFIN(CATLFN,THBUF,224,IEOF)
      .  .  CALL DSTAT(CATLFN,ISTS,CRA)
      .  .  EXIT LOOP IF(IEOF.GE.3)
      .  .  DECODE(9,1001,THBUF) ITAP
1001  .  .  FORMAT(3X,I6)
      .  .  DECODE(3,1002,THBUF(160)) IDASH
1002  .  .  FORMAT(I3)
      .  .  IF(ITAP.EQ.JTAPE.AND.IDSH.EQ.0)
      .  .  .  GOTO 2
      .  .  END IF
      .  .  IF(ITAP.EQ.JTAPE.AND.IDSH.EQ.IDASH)
2     .  .  .  CRA=CRA-2
      .  .  .  WRITE(3,20) ITAP,IDASH
      .  .  .  WRITE(LFN,20) ITAP,IDASH
20    .  .  .  FORMAT(/' Tape #',I6,'-',I1)
      .  .  .  FNDFLG=1
      .  .  .  CALL CATED(CRA)
      .  .  .  CALL CATMM(CRA)
      .  .  END IF
      .  END LOOP
      .  K=K+1
      .  KTN=KTN-1
      .  IF(IEOF.GE.3.AND.FNDFLG.EQ.0)
      .  .  WRITE(3,40) JTAPE,IDSH
      .  .  WRITE(LFN,40) JTAPE,IDSH
40    .  .  FORMAT(' ***ERROR: TAPE NUMBER',I6,'-',I3,'  NOT FOUND')
      .  END IF
      .  ITEM=ITEM+1
      END WHILE
      STOP
      END

      SUBROUTINE CATED(CRA)
C Subroutine used with program ROSED to edit the standard ROSE
C header part of the Catalog tape header record.
C
C      LAST MODIFIED 4/26/82 BY S. LATRAILLE
```

```
C **** NOTE: ADDED A LINE (AFTER CALL TO DSTAT TO DETERMINE
C      POSITION IN CATALOG AFTER WRITING) TO
C      REPOSITION BACK TO BEGINNING OF CURRENT RECORD
C
      COMMON/BUFER/ THBUF,CATLFN,LFN
      INTEGER CRA,THBUF(224),RDATE(3),ADATE(3),UDATE(3),CDATE(3)
      INTEGER TRAY(7),CATLFN,IDATE(3)
      COMMON/ITCM/TRAY,CSEC,JULD
      INTEGER*6 TDST,TDET,CSEC,ISTS
      DIMENSION PLACE(5),DNAME(9),SHTLN(10),EXCODE(4)
C
C
      CALL DATE(IDATE)
      DECODE(84,7000,THBUF) JTYPE,ITAP,NSLN,PLACE,IDOC,
     +RDATE,ADATE,UDATE,CDATE
7000  FORMAT(I3,2I6,5A6,13A3)
      DECODE(174,7010,THBUF(95)) IID,DNAME,EXCODE,NFILES,
     +TDST,TDET,SHTLN
7010  FORMAT(I4,12A6,A2,I6,I14,1X,I14,1X,10A6)
      DECODE(3,7011,THBUF(160)) IDASH
7011  FORMAT(I3)
1     WRITE(3,1000)
1000  FORMAT(' Enter the number of the variable you want to change.',
     +/' Only one can be entered at a time, but program will loop',
     +' to do more',
     +/' If ALL, enter 0; if PAU, enter 99:')
      WRITE(3,1001)
1001  FORMAT(//' 1 - Tape No.            5 - Date rcvd.        9 - No.',
     +' of files',
     +/' 2 - No. Shotline #s     6 - Date archived    10 - Tape data',
     +' start time',
     +/' 3 - Inst. rcvd from     7 - Instrument I.D.  11 - Tape data',
     +' end time',
     +/' 4 - Document. code      8 - Designer name    12 - Shotline',
     +' designation',/' 13 - Dash no.(position on archive tape')
      READ(3,) KGOTO
      IF(KGOTO.EQ.0) GO TO 50
      IF(KGOTO.EQ.99) GO TO 99
      GO TO (50,2,3,4,5,6,7,8,9,10,11,12,13),KGOTO
50    WRITE(3,2001) ITAP
2001  FORMAT(' Archive tape no.',I4,'  Enter tape no.:')
      READ(0,) ITAP
      IF(KGOTO) 99,,99
2     WRITE(3,2002) NSLN
2002  FORMAT(' Number of shotline #s:',I6,'  Enter number:')
      READ(0,) NSLN
      IF(KGOTO) 99,,99
3     WRITE(3,2003) PLACE
2003  FORMAT(' Institute recd from:',5A6,/' Enter institute (30 char):')
      READ(0,3003) PLACE
3003  FORMAT(5A6)
      IF(KGOTO) 99,,99
4     WRITE(3,2004) IDOC
2004  FORMAT(' Documentation code:',A3,' Enter doc. code:')
      READ(0,3004) IDOC
3004  FORMAT(A3)
      IF(KGOTO) 99,,99
5     WRITE(3,2005) RDATE
2005  FORMAT(' Date Arch. tape recd:',3A3,/' Enter date: ')
      READ(3,3005) RDATE
3005  FORMAT(3A3)
      IF(KGOTO) 99,,99
6     WRITE(3,2006) ADATE
2006  FORMAT('  Date archived:',3A3,/' Enter date:')
      READ(0,3005) ADATE
      IF(KGOTO) 99,,99
7     WRITE(3,2007) IID
2007  FORMAT(' Instrument ID#: ',I6,' Enter ID #:')
      READ(0,) IID
      IF(KGOTO) 99,,99
8     WRITE(3,2008) DNAME
2008  FORMAT(' Designers name & address:',9A6,/' Enter name & address:')
      READ(0,3008) DNAME
3008  FORMAT(9A6)
      IF(KGOTO) 99,,99
9     WRITE(3,2009) NFILES
2009  FORMAT(' Number of files (events) on tape: ',I6,
     +/' Enter number of files:')
      READ(0,) NFILES
      IF(KGOTO) 99,,99
```

```
160: 10    CSEC=TDST
161:       CALL CNTITM
162:       WRITE(3,2010) (TRAY(I),I=1,5)
163: 2010  FORMAT(' Tape data start time: ',5I4,/' Enter data start time:')
164:       READ(0,) (TRAY(I),I=1,5)
165:       CALL ITMCNT
166:       TDST=CSEC
167:       IF(KGOTO) 99,,99
168: 11    CSEC=TDET
169:       CALL CNTITM
170:       WRITE(3,2011) (TRAY(I),I=1,5)
171: 2011  FORMAT(' Tape data end time: ',5I4,/' Enter data end time:')
172:       READ(0,) (TRAY(I),I=1,5)
173:       CALL ITMCNT
174:       TDET=CSEC
175:       IF(KGOTO) 99,,99
176: 12    WRITE(3,2012) SHTLN
177: 2012  FORMAT(' Shotline designation(s):',10A6,/' Enter shotline(s) - ',
178:      +'60 char:')
179:       READ(0,3012) SHTLN
180: 3012  FORMAT(10A6)
181:       IF(KGOTO) 99,,99
182: 13    WRITE(3,2013) IDASH
183: 2013  FORMAT(' Dash no.,or position on tape:',I3,/' Enter dash no.:')
184:       READ(0,) IDASH
185: 99    WRITE(3,2015)
186: 2015  FORMAT(' PAU WITH EDIT? YES=1:')
187:       READ(0,) IEND
188:       IF(IEND.NE.1) GO TO 1
189:       FOR I=1,3
190:       .  UDATE(I)=IDATE(I)
191:       .  CDATE(I)=IDATE(I)
192:       END FOR
193: C ENCODE CHANGED AS WELL AS OLD VALUES INTO THBUF
194:       ENCODE(84,7000,THBUF) JTYPE,ITAP,NSLN,PLACE,IDOC,
195:      +RDATE,ADATE,UDATE,CDATE
196:       ENCODE(174,7010,THBUF(95)) IID,DNAME,EXCODE,NFILES,
197:      +TDST,TDET,SHTLN
198:       ENCODE(3,7011,THBUF(160)) IDASH
199: C POSITION TO CURRENT RECORD AND REWRITE IT
200:       CALL DPOS(CATLFN,CRA)
201:       CALL BUFOUT(CATLFN,THBUF,224,IEOF)
202:       CALL DPOS(CATLFN,CRA)
203:       CALL DSTAT(CATLFN,ISTS,CRA)
204:       WRITE(LFN,6000) ITAP,IDASH,IDATE,KGOTO
205: 6000  FORMAT(' CATALOG FILE HEADER FOR ARCHIVE TAPE #',I6,
206:      +'-',I3,'  EDITED',
207:      +' AND REWRITTEN TO REVCAT ON ',3A3,' - LAST CHANGE #',I3)
208: 999   RETURN
209:       END

210: C*****************************************************************
211:       SUBROUTINE CATMM(CRA)
212: C
213: C   Subroutine to edit the HIG added part of the header
214: C
215: C      LAST MODIFIED 5/13/82 S.LATRAILLE
216: C
217:       COMMON/BUFER/ THBUF,CATLFN,LFN
218:       INTEGER CRA,THBUF(224),ENUMMX,ENUMMN,DEPEMN,DEPEMX,EDEPMN
219:       INTEGER TRAY(7),CATLFN,EDEPMX,WDEPI,IDATE(3)
220:       REAL ILAT,ILON
221:       COMMON/ITCM/TRAY,CSEC,JULD
222:       INTEGER*6 CSEC,ISTS,DSTMAX,DSTMIN
223: C
224: C
225:       CALL DATE(IDATE)
226:       DECODE(9,5999,THBUF) ITAP
227: 5999  FORMAT(3X,I6)
228:       DECODE(96,6000,THBUF(29)) INUM,ENUMMX,ENUMMN,DSTMAX,
229:      +DSTMIN,JXPLMX,JXPLMN,WDEPI,DEPEMX,DEPEMN,
230:      +IDEP,EDEPMX,EDEPMN,ICHNMX,ICHNMN,JTYPMX,JTYPMN
231: 6000  FORMAT(3I6,2I13,2I2,6I6,2I3,2I2,2X)
232:       DECODE(102,6001,THBUF(61)) SIZEMX,SIZEMN,RANGMX,RANGMN,
233:      +ILAT,ILON,ELATMX,ELATMN,ELONMX,ELONMN
234: 6001  FORMAT(10F10.4,2X)
235: C
236: 1     WRITE(3,1000)
237: 1000  FORMAT(' Enter the number of the variable you want to change.',
238:      +/' Only one can be entered at a time, but program will loop',
239:      +' to do more',
240:      +/' If ALL, enter 0; if PAU, enter 99:')
```

```
241:       WRITE(3,1001)
242: 1001  FORMAT(//' 1 - IID No.            8 - Max Shot Depth   15 - Max',
243:      +' range',
244:      +/' 2 - Min Shot #          9 - Water dep recv   16 - Inst',
245:      +' latitude',
246:      +/' 3 - Max Shot #         10 - Min WD at shot   17 - Inst',
247:      +' longitude',
248:      +/' 4 - Min Start Time     11 - Max WD at shot   18 - Min shot',
249:      +' latitude',
250:      +/' 5 - Max Start Time     12 - Min shot size    19 - Max shot',
251:      +' latitude',
252:      +/' 6 - Recv Depth         13 - Max shot size    20 - Min shot',
253:      +' longitude',
254:      +/' 7 - Min Shot Depth     14 - Min range        21 - Max shot',
255:      +' longitude',
256:      +/T48,'22 - Event header addr')
257:       READ(3,) KGO
258:       IF(KGO.EQ.0) GO TO 50
259:       IF(KGO.EQ.22) GO TO 22
260:       IF(KGO.EQ.99) GO TO 99
261:       GO TO (50,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21),KGO
262: 50    WRITE(3,2001) INUM
263: 2001  FORMAT(' Instrument I.D. no.',I4,'  Enter I.D. no.:')
264:       READ(0,) INUM
265:       IF(KGO) 99,,99
266: 2     WRITE(3,2002) ENUMMN
267: 2002  FORMAT(' Min shot #:',I6,'  Enter number:')
268:       READ(0,) ENUMMN
269:       IF(KGO) 99,,99
270: 3     WRITE(3,2003) ENUMMX
271: 2003  FORMAT(' Max shot #:',I6,'  Enter number:')
272:       READ(0,) ENUMMX
273:       IF(KGO) 99,,99
274: 4     CSEC=DSTMIN
275:       CALL CNTITM
276:       WRITE(3,2010) (TRAY(I),I=1,5)
277: 2010  FORMAT(' Min data start time: ',5I4,/' Enter data start time:')
278:       READ(0,) (TRAY(I),I=1,5)
279:       CALL ITMCNT
280:       DSTMIN=CSEC
281:       IF(KGO) 99,,99
282: 5     CSEC=DSTMAX
283:       CALL CNTITM
284:       WRITE(3,2011) (TRAY(I),I=1,5)
285: 2011  FORMAT(' Max data start time: ',5I4,/' Enter data start time:')
286:       READ(0,) (TRAY(I),I=1,5)
287:       CALL ITMCNT
288:       DSTMAX=CSEC
289:       IF(KGO) 99,,99
290: 6     WRITE(3,2004) IDEP
291: 2004  FORMAT(' Recv depth:',I6,' Enter recv depth:')
292:       READ(0,) IDEP
293:       IF(KGO) 99,,99
294: 7     WRITE(3,2005) EDEPMN
295: 2005  FORMAT(' Min event depth:',I6,/' Enter depth: ')
296:       READ(3,) EDEPMN
297:       IF(KGO) 99,,99
298: 8     WRITE(3,2006) EDEPMX
299: 2006  FORMAT('  Max event depth:',I6,/' Enter depth:')
300:       READ(0,) EDEPMX
301:       IF(KGO) 99,,99
302: 9     WRITE(3,2007) WDEPI
303: 2007  FORMAT(' Water depth, recv: ',I6,' Enter water depth:')
304:       READ(0,) WDEPI
305:       IF(KGO) 99,,99
306: 10    WRITE(3,2008) DEPEMN
307: 2008  FORMAT(' Min water depth at event:',I6,/' Enter depth:')
308:       READ(0,) DEPEMN
309:       IF(KGO) 99,,99
310: 11    WRITE(3,2009) DEPEMX
311: 2009  FORMAT(' Max water depth at event: ',I6,
312:      +/' Enter depth:')
313:       READ(0,) DEPEMX
314:       IF(KGO) 99,,99
315: 12    WRITE(3,2012) SIZEMN
316: 2012  FORMAT(' Min shot size:',F10.4,/' Enter size:')
317:       READ(0,) SIZEMN
318:       IF(KGO) 99,,99
319: 13    WRITE(3,2013) SIZEMX
320: 2013  FORMAT(' Max shot size:',F10.4,' Enter size:')
```

```
      READ(0,) SIZEMX
      IF(KGO) 99,,99
14    WRITE(3,2014) RANGMN
2014  FORMAT(' Min range:',F10.4,' Enter range:')
      READ(0,) RANGMN
      IF(KGO) 99,,99
15    WRITE(3,2015) RANGMX
2015  FORMAT(' Max range:',F10.4,' Enter range:')
      READ(0,) RANGMX
      IF(KGO) 99,,99
16    WRITE(3,2016) ILAT
2016  FORMAT(' Instrument latitude:',F10.4,' Enter latitude:')
      READ(0,) ILAT
      IF(KGO) 99,,99
17    WRITE(3,2017) ILON
2017  FORMAT(' Instrument longitude:',F10.4,' Enter longitude:')
      READ(0,) ILON
      IF(KGO) 99,,99
18    WRITE(3,2018) ELATMN
2018  FORMAT(' Min event latitude:',F10.4,' Enter latitude:')
      READ(0,) ELATMN
      IF(KGO) 99,,99
19    WRITE(3,2019) ELATMX
2019  FORMAT(' Max event latitude:',F10.4,' Enter latitude:')
      READ(0,) ELATMX
      IF(KGO) 99,,99
20    WRITE(3,2020) ELONMN
2020  FORMAT(' Min event longitude:',F10.4,' Enter longitude:')
      READ(0,) ELONMN
      IF(KGO) 99,,99
21    WRITE(3,2021) ELONMX
2021  FORMAT(' Max event longitude:',F10.4,' Enter longitude:')
      READ(0,) ELONMX
      IF(KGO) 99,,99
22    IDADR=THBUF(153)
      WRITE(3,2022) IDADR
2022  FORMAT(' Event header start address:',I8,' Enter address:')
      READ(0,)IDADR
      THBUF(153)=IDADR
99    WRITE(3,2099)
2099  FORMAT(' PAU WITH EDIT? YES=1:')
      READ(0,) IEND
      IF(IEND.NE.1) GO TO 1
      ENCODE(96,6000,THBUF(29)) INUM,ENUMMX,ENUMMN,DSTMAX,
     +DSTMIN,JXPLMX,JXPLMN,WDEPI,DEPEMX,DEPEMN,
     +IDEP,EDEPMX,EDEPMN,ICHNMX,ICHNMN,JTYPMX,JTYPMN
      ENCODE(102,6001,THBUF(61)) SIZEMX,SIZEMN,RANGMX,RANGMN,
     +ILAT,ILON,ELATMX,ELATMN,ELONMX,ELONMN
      CALL DPOS(CATLFN,CRA)
      CALL BUFOUT(CATLFN,THBUF,224,IEOF)
      CALL DSTAT(CATLFN,ISTS,CRA)
      WRITE(LFN,8000) ITAP,IDATE,KGO
8000  FORMAT(' CATALOG FILE HEADER FOR ARCHIVE TAPE #',I6,' EDITED',
     +' AND REWRITTEN TO REVCAT ON ',3A3,' - LAST CHANGE #',I3)
999   RETURN
      END
```

```
 1: C PROGRAM  A D H E D  TO ADD HEADERS BACK ONTO CATALOG FILE AFTER A
 2: C PROGRAM ABORT OR SYSTEM FAILURE
 3: C
 4: C        LAST MODIFIED 5/20/81
 5: C              AS 20=THEAD, WHICH MUST CONTAIN ALL TAPE HEADERS FROM
 6: C                      THE CATALOG FILE, AS 30=REVCAT, WHICH IS MINUS
 7: C                      ITS HEADERS
 8: C                     AS 20=THEAD,30=REVCAT
 9: C
10:       INTEGER CATBUF(112),THBUF(224)
11:       OPEN 20
12:       OPEN 30
13:       LOOP(3)
14:       .  CALL BUFIN(30,CATBUF,112,IEOF)
15:       END LOOP
16:       JHRA=CATBUF(112)
17:       WRITE(3,100) JHRA
18: 100   FORMAT(1X,'NOW POSITION TO REC',I5,' TO WRITE HEADERS')
19:       CALL DPOS(30,JHRA)
20:       LOOP
21:       .  CALL BUFIN(20,THBUF,224,IEOF)
22:       .  EXIT LOOP IF (IEOF.GE.3)
23:       .  CALL BUFOUT(30,THBUF,224,IEOF)
24:       END LOOP
25:       ENDFILE 30
26:       STOP
27:       END
```

```
 1: C PROGRAM ADTAPE TO OPTIONALLY (1) ADD BACK TAPE HEADERS TO A
 2: C CATALOG FILE WITH DATA RECORDS ONLY( THIS COULD HAPPEN IF
 3: C SYSTEM CRASHES DURING AN ARCHIVE OR IF PROGRAM ABORTS DURING
 4: C ARCHIVE), OR (2) TO ADD A NEW TAPE HEADER AND NEW DATA RECORDS
 5: C TO A COMPLETE CATALOG FILE.
 6: C
 7: C               USE THIS PROGRAM WITH MACRO DUPCAT
 8: C               LAST MODIFIED 1/13/81 SLL
 9: C ASSIGN 30=CATALOG TO WRITE TO,11=THEAD,40=INPUT CATALOG FILE FOR OPT 2
10: C          (UNBLKD)  (BLKD)   (UNBLKD)
11:       INTEGER CATBUF(112),THBUF(224),THBUF1(224)
12:       INTEGER*6 ISTAT
13:       OPEN 30
14:       OPEN 11
15:       OPEN 40
16:       WRITE(3,98)
17: 98    FORMAT(' PROGRAM ADTAPE TO OPTIONALLY (1) ADD BACK TAPE HEADERS',
18:      +/' TO A CATALOG FILE WITH DATA RECORDS ONLY( THIS COULD HAPPEN',
19:      +/' IF SYSTEM CRASHES DURING AN ARCHIVE OR IF PROGRAM ABORTS ',
20:      +/' DURING ARCHIVE), OR (2) TO ADD A NEW TAPE HEADER AND NEW ',
21:      +/' EVENT HEADER RECORDS TO A COMPLETE CATALOG FILE.',
22:      +//' ASSIGN 30=REVCAT,11=THEAD,40=INPUT CATFILE FOR OPTION 2')
23:       WRITE(3,100)
24: 100   FORMAT(/' OPTION 1 (ADD BACK TAPE HDRS);'
25:      +/' OR OPTION 2 (ADD NEW TAPE HEADER & DATA RECORDS)')
26:       READ(0,)IOPT
27: 1     CALL DPOS(30,2)
28:       CALL BUFIN(30,CATBUF,112,IEOF)
29:       KRA=CATBUF(112)
30:       CALL DPOS(30,KRA)
31:       IF(IOPT.EQ.2) GO TO 10
32:       LOOP
33:       .  BUFFER IN(11,THBUF,B,224,MS,MLEN)
34:       .  CALL STATUS(11)
35:       .  EXIT LOOP IF(MS.GE.3)
36:       .  CALL BUFOUT(30,THBUF,224,IEOF)
37:       END LOOP
38:       ENDFILE 30
39:       STOP HEAD
40: 10    LOOP
41: C SAVE OFF TAPE HEADERS
42:       .  CALL BUFIN(30,THBUF,224,IEOF)
43:       .  EXIT LOOP IF(IEOF.GE.3)
44:       .  BUFFER OUT(11,THBUF,B,224,MS,MLEN)
45:       END LOOP
46:       ENDFILE 11
47: C POSITION CATALOG FILE TO OLD HEADER START ADDRESS
48: C THIS ADDRESS IS NOW THE START ADDR OF NEW DATA RECORDS
49:       CALL DPOS(30,KRA)
50:       WRITE(3,103) KRA
51: 103   FORMAT(' NEW DATA ADDRESS:',I7)
52:       WRITE(3,101)
53: 101   FORMAT(' ENTER TAPE # AND DASH # TO BE ADDED:')
54:       READ(0,) NTAPE,JFN
55:       IDASH=JFN
56:       CALL DPOS(40,2)
57:       CALL BUFIN(40,CATBUF,112,IEOF)
58:       CALL DPOS(40,CATBUF(112))
59: C SEARCH SECOND CAT FILE FOR SPECIFIED TAPE
60:       LOOP
61: 2     .  CALL BUFIN(40,THBUF1,224,IEOF)
62:       .  DECODE(9,1000,THBUF1) ITAP
63: 1000  .  FORMAT(3X,I6)
64:       .  DECODE(6,1001,THBUF1(121)) NFILES
65: 1001  .  FORMAT(I6)
66:       .  IF(IEOF.GE.3)
67:       .  .  WRITE(3,102)
68: 102   .  .  FORMAT(' TAPE SPECIFIED NOT IN FILE 40')
69:       .  .  STOP FILE
70:       .  END IF
71:       .  IF (ITAP.EQ.NTAPE)
72:       .  .  IF(JFN.NE.1)
73:       .  .  .  JFN=JFN-1
74:       .  .  .  GO TO 2
75:       .  .  END IF
76:       .  .  EXIT LOOP
77:       .  END IF
78:       END LOOP
79: C POSITION TO NEW DATA RECORDS
80:       IADD=THBUF1(153)
```

```
 81: C       WRITE(3,110) THBUF1(153)
 82: C110    FORMAT(' DATA ADD IN HDR=',I6,' ENTER DATA ADD:')
 83: C       READ(0,) IADD
 84:        CALL DPOS(40,IADD)
 85:        THBUF1(153)=KRA
 86:        LOOP(NFILES)
 87:        .  CALL BUFIN(40,CATBUF,112,IEOF)
 88:        .  EXIT LOOP IF(IEOF.GE.3)
 89:        .  CALL BUFOUT(30,CATBUF,112,IEOF)
 90:        END LOOP
 91: C DETERMINE NEW HEADER START ADDRESS
 92:        CALL DSTAT(30,ISTAT,KRA)
 93:        WRITE(3,120) KRA
 94: 120    FORMAT(' NEW HDR START ADDRESS IS:',I6)
 95:        REWIND 11
 96:        LOOP
 97:        .  BUFFER IN(11,THBUF,B,224,MS,MLEN)
 98:        .  CALL STATUS(11)
 99:        .  EXIT LOOP IF(MS.GE.3)
100:        .  CALL BUFOUT(30,THBUF,224,IEOF)
101:        END LOOP
102:        CALL DSTAT(30,ISTAT,NRA)
103: C WRITE HEADER FOR NEW TAPE TO CATALOG
104:        CALL BUFOUT(30,THBUF1,224,IEOF)
105:        ENDFILE 30
106:        CALL DPOS(30,1)
107:        CALL BUFIN(30,CATBUF,112,IEOF)
108:        CATBUF(112)=KRA
109:        CALL BUFOUT(30,CATBUF,112,IEOF)
110:        REWIND 11
111:        WRITE(3,130)
112: 130    FORMAT(' ADD ANOTHER TAPE? YES=1')
113:        READ(0,) IGO
114:        IF(IGO.EQ.1) GO TO 1
115:        STOP
116:        END
```

```
C PROGRAM DELHDR TO DELETE TAPE HEADERS FROM
C THE CATALOG FILE
C
C               LAST MODIFIED 5/13/82 SLL
C ASSIGN 30=CATALOG ,11=FILE TO SAVE OFF HEADERS
C          (UNBLKD)  (BLKD)
      INTEGER CATBUF(112),THBUF(224),IDATE(3)
      OPEN 30
      OPEN 11
      WRITE(3,100)
100   FORMAT(/' ENTER TAPE AND DASH NUMBER'
     +/' TO BE DELETED. ')
      READ(0,) NTAPE,IDASH
      CALL DPOS(30,2)
      CALL BUFIN(30,CATBUF,112,IEOF)
      KRA=CATBUF(112)
      CALL DPOS(30,KRA)
      LOOP
C SAVE OFF TAPE HEADERS
2     . CALL BUFIN(30,THBUF,224,IEOF)
      . IF(IEOF.GE.3) GO TO 5
      . DECODE(9,4000,THBUF) ITAP
4000  . FORMAT(3X,I6)
      . DECODE(3,4001,THBUF(160)) IDSH
4001  . FORMAT(I3)
      . IF(ITAP.NE.NTAPE) GO TO 3
      . IF(IDASH.NE.IDSH) GOTO 3
      . DECODE(9,4002,THBUF(20)) (IDATE(I),I=1,3)
4002  . FORMAT(3A3)
      . WRITE(3,102) ITAP,IDSH,IDATE
102   . FORMAT(' DELETE TAPE # ',I6,' DASH ',I3,' ARCHIVED ON',
     +. 3X,3A3,' ? IF YES ENTER 1:')
      . READ(0,) IDD
      . IF(IDD.NE.1)
      . . GO TO 3
      . ELSE
      . . GO TO 2
      . END IF
3     . BUFFER OUT(11,THBUF,B,224,MS,MLEN)
      END LOOP
5     ENDFILE 11
C POSITION CATALOG FILE TO START ADDRESS OF HEADER RECORDS
      CALL DPOS(30,KRA)
      REWIND 11
      LOOP
      . BUFFER IN(11,THBUF,B,224,MS,MLEN)
      . CALL STATUS(11)
      . EXIT LOOP IF(MS.GE.3)
      . CALL BUFOUT(30,THBUF,224,IEOF)
      END LOOP
      ENDFILE 30
      STOP
      END
```

```
 1: C PROGRAM DELTAP TO DELETE THE LAST TAPE WHICH WAS ARCHIVED
 2: C FROM THE CATALOG FILE ALONG WITH ITS DATA RECORDS
 3: C
 4: C               LAST MODIFIED 9/28/81 SLL
 5: C ASSIGN 30=CATALOG ,11=THEAD TO SAVE OFF HEADERS
 6: C          (UNBLKD)  (BLKD)
 7:       INTEGER CATBUF(112),THBUF(224),IDATE(3)
 8:       OPEN 30
 9:       OPEN 11
10:       WRITE(3,100)
11: 100   FORMAT(/' ENTER TAPE AND DASH NUMBER'
12:      +/' TO BE DELETED. THIS SHOULD BE THE LAST ONE ARCHIVED:')
13:       READ(0,) NTAPE,JFN
14:       IDASH=JFN
15:       CALL DPOS(30,2)
16:       CALL BUFIN(30,CATBUF,112,IEOF)
17:       KRA=CATBUF(112)
18:       CALL DPOS(30,KRA)
19:       LOOP
20: C SAVE OFF TAPE HEADERS
21:       .  CALL BUFIN(30,THBUF,224,IEOF)
22:       .  IF(IEOF.GE.3) STOP ERROR
23:       .  DECODE(9,4000,THBUF) ITAP
24: 4000  .  FORMAT(3X,I6)
25:       .  IF(ITAP.EQ.NTAPE)
26:       .  .  IF(JFN.NE.1)
27:       .  .  .  JFN=JFN-1
28:       .  .  .  GO TO 3
29:       .  .  END IF
30:       .  .  IDADR=THBUF(153)
31:       .  .  DECODE(9,4001,THBUF(20)) (IDATE(I),I=1,3)
32: 4001  .  .  FORMAT(3A3)
33:       .  .  WRITE(3,102) ITAP,IDASH,IDATE
34: 102   .  .  FORMAT(' DELETING TAPE # ',I6,' DASH ',I3,' ARCHIVED ON',
35:      +.  .  3X,3A3)
36:       .  .  GO TO 5
37:       .  END IF
38:  3    .  BUFFER OUT(11,THBUF,B,224,MS,MLEN)
39:       END LOOP
40:  5    ENDFILE 11
41: C POSITION CATALOG FILE TO OLD DATA START ADDRESS
42: C THIS ADDRESS IS NOW THE START ADDR OF HEADER RECORDS
43:       CALL DPOS(30,IDADR)
44:       WRITE(3,103) IDADR
45: 103   FORMAT(' NEW HEADER ADDRESS:',I7)
46:       REWIND 11
47:       LOOP
48:       .  BUFFER IN(11,THBUF,B,224,MS,MLEN)
49:       .  CALL STATUS(11)
50:       .  EXIT LOOP IF(MS.GE.3)
51:       .  CALL BUFOUT(30,THBUF,224,IEOF)
52:       END LOOP
53:       ENDFILE 30
54:       CALL DPOS(30,1)
55:       CALL BUFIN(30,CATBUF,112,IEOF)
56:       CATBUF(112)=IDADR
57:       CALL BUFOUT(30,CATBUF,112,IEOF)
58:       STOP
59:       END
```

```
 1: C PROGRAM BULLETIN TO WRITE BULLETIN FROM SORTED BINARY CATALOG FILE
 2: C   LAST MODIFIED 2/21/81
 3:       NAME BULLETIN
 4:       INTEGER TRAY(7),TIME(5),CATBUF(112),PTAP,PINUM,BENUM,ENUM,CRA
 5:       INTEGER CATLFN,RLFN,FTAP,FINUM,FENUM
 6:       INTEGER*6 CSEC,BTIME,DST,ISTAT,FTIME,ORIGIN
 7:       EQUIVALENCE (CATBUF(1),ITYPE),(CATBUF(2),ITAP),(CATBUF(3),INUM)
 8:       EQUIVALENCE (CATBUF(4),ENUM),(CATBUF(5),DST),(CATBUF(67),SHTLN)
 9:       COMMON/ITCM/TRAY,CSEC,JULD
10:       DATA CATLFN/30/,RLFN/10/
11: C
12: C
13:       OPEN CATLFN
14:       OPEN RLFN
15: C      WRITE(3,100)
16: C100   FORMAT(' ENTER START RECORD ADDRESS:')
17: C      READ(0,) ICRA
18:       ICRA=0
19:       ICOUNT=0
20:       ISUB=0
21:       ITOT=0
22:       CALL DPOS(CATLFN,ICRA)
23:       CALL BUFIN(CATLFN,CATBUF,112,IEOF)
24:       FSLN=SHTLN
25:       CALL DPOS(CATLFN,ICRA)
26:       WRITE(RLFN,5001) FSLN
27: 5001  FORMAT(///'  SHOTLINE DESIGNATION: ',A6)
28:       WRITE(RLFN,5002)
29: 5002  FORMAT(//'      Event #s      Num      Instrument           ',
30:      +'Beginning and Ending           Archive')
31:       WRITE(RLFN,5003)
32: 5003  FORMAT('      Included      Evnts   I.D. (Origin)          Data ',
33:      +'Start Times                Tape No.')
34:       LOOP
35: 1     . CALL BUFIN(CATLFN,CATBUF,112,IEOF)
36:       . CALL DSTAT(CATLFN,ISTAT,CRA)
37:       . ICOUNT=ICOUNT+1
38:       . ITOT=ITOT+1
39:       . EXIT LOOP IF(IEOF.GE.3)
40:       . IF(ITYPE.NE.1)
41:       . . ICOUNT=ICOUNT-1
42:       . . GO TO 1
43:       . END IF
44:       . IF(ICOUNT.EQ.1)
45:       . . PSLN=SHTLN
46:       . . PSLN=SHTLN
47:       . . FTAP=ITAP
48:       . . FINUM=INUM
49:       . . FENUM=ENUM
50:       . . FTIME=DST
51:       . ELSE IF(ICOUNT.NE.1)
52:       . . IF(INUM.NE.PINUM.OR.SHTLN.NE.PSLN.OR.ITAP.NE.PTAP)
53: C CONVERT TIMES TO INTEGER ARRAYS
54:       . . . CSEC=FTIME
55:       . . . CALL CNTITM
56:       . . . FOR I=1,5
57:       . . . . TIME(I)=TRAY(I)
58:       . . . END FOR
59:       . . . CSEC=BTIME
60:       . . . CALL CNTITM
61:       . . . CALL ORID(PINUM,ORIGIN)
62:       . . . ICT=ICOUNT-1
63:       . . . ITOT=ITOT-1
64:       . . . WRITE(RLFN,5010) FENUM,BENUM,ICT,PINUM,ORIGIN,(TIME(I),I=1,5
65:      +. . . (TRAY(I),I=2,5),PTAP
66: 5010  . . . FORMAT(1X,I6,' to',I6,3X,I3,3X,I6,1X,A6,5X,I4,I2,'/',I2,1X,I
67:      +. . . I2,' to ',I2,'/',I2,1X,I2,':',I2,6X,I6)
68:       . . . ISUB=ISUB+ICT
69:       . . . ICOUNT=0
70:       . . . CRA=CRA-1
71:       . . . CALL DPOS(CATLFN,CRA)
72:       . . END IF
73:       . END IF
74:       . IF(SHTLN.NE.PSLN)
75:       . . WRITE(RLFN,5013) ISUB
76: 5013  . . FORMAT(/'      Subtotal',I6)
77:       . . WRITE(RLFN,5001) SHTLN
78:       . . WRITE(RLFN,5002)
79:       . . WRITE(RLFN,5003)
80:       . . ISUB=0
```

```
      . END IF
      . PSLN=SHTLN
      . PTAP=ITAP
      . PINUM=INUM
      . BENUM=ENUM
      . BTIME=DST
      END LOOP
      CALL ORID(PINUM,ORIGIN)
      WRITE(RLFN,5010) FENUM,BENUM,ICT,PINUM,ORIGIN,(TIME(I),I=1,5),
     +(TRAY(I),I=2,5),PTAP
      WRITE(RLFN,5013) ISUB
      WRITE(RLFN,5011) ITOT
5011  FORMAT(///' TOTAL NUMBER OF EVENTS PROCESSED= ',I9)
      STOP BULL
      END
      SUBROUTINE ORID(IID,ORIGIN)
      INTEGER*6 OR(12),ORIGIN
      DATA OR(1)/'WHOI  '/,OR(2)/'LDGO  '/,OR(3)/'UTMSI '/
      DATA OR(4)/'SCRIPP'/,OR(5)/'UW    '/,OR(6)/'HIG   '/
      DATA OR(7)/'OSU   '/,OR(8)/'MIT   '/,OR(9)/'UCSB  '/
      DATA OR(10)/'NORDA '/,OR(11)/'MIT   '/,OR(12)/'NRL   '/
      RID=FLOAT(IID)/100.
      J=IFIX(RID)
      C=FLOAT(J)
      B=RID-C
      IF(B.GT..001)
      . J=J+1
      END IF
      ORIGIN=OR(J)
      RETURN
      END
```

```
 1: C PROGRAM SEARCH:3  TO SEARCH THE EVENT CATALOG AND LIST WHERE TO
 2: C FIND REQUESTED RECORDS
 3: C
 4: C :::::::::::::::::::::::::: VERS 3 - WRITE BINARY HEADERS ::::::::::::
 5: C**************************************************************************
 6: C    WRITTEN BY SHARON LATRAILLE  HIG 363 X7796   LAST UPDATED 9/27/82
 7: C
 8: C     ICHN=# OF CHANNELS
 9: C     CCODE=CODE FOR DUPLICATE COMPONENT PARAMETERS
10: C           (IF 1, ONLY COMPONENT 1 IS CODED)
11: C     COMBUF=CHANNEL CODE BUFFER - CHANNELS 1 THRU ICHN ARE CODED WITH
12: C            CHANNEL TYPE (1 THRU 8) AND THE REST, ICHN+1 TO 24 ARE =0.
13: C            IF CCODE=1, CHANNELS 2 THRU ICHN ARE SAME AS CHANNEL 1.
14: C
15:       NAME SEARCH
16:       INTEGER COMBUF(24),CATBUF(112),JBUF(112),CCODE,INUM,ENUM
17:       INTEGER CATLFN,TAPENM,ICSEL(8),CHSEL(24),ISTAT(2),FNUM
18:       INTEGER RERR,ELOER,BUB,SAMP,TRAY(7),FHDR(256),TYPE,HDLFN
19:       INTEGER WDEPI,WDEPE,IDEP,EDEP,ICHN,EXPL
20:       REAL ILAT,ILON
21:       COMMON /ITCM/TRAY,CSEC,JULD
22:       COMMON /SEL/ JBUF
23:       EQUIVALENCE (JBUF(1),CATBUF(1))
24:       EQUIVALENCE (CATBUF(1),JTYPE),(CATBUF(2),TAPENM),(CATBUF(3),INUM)
25:       EQUIVALENCE (CATBUF(4),ENUM),(CATBUF(5),DST),(CATBUF(7),SBT)
26:       EQUIVALENCE (CATBUF(9),SIZE),(CATBUF(11),RANGE)
27:       EQUIVALENCE (CATBUF(13),ILAT),(CATBUF(15),ILON),(CATBUF(17),ELAT)
28:       EQUIVALENCE (CATBUF(19),ELON),(CATBUF(21),EXPL),(CATBUF(22),WDEPI)
29:       EQUIVALENCE (CATBUF(23),WDEPE),(CATBUF(24),IDEP),(CATBUF(25),EDEP)
30:       EQUIVALENCE (CATBUF(26),ICHN),(CATBUF(27),TYPE)
31:       EQUIVALENCE (CATBUF(28),RERR),(CATBUF(29),ELOER),(CATBUF(30),BUB)
32:       EQUIVALENCE (CATBUF(31),SAMP),(CATBUF(32),NWDS)
33:       EQUIVALENCE (CATBUF(33),FNUM),(CATBUF(34),NRC),(CATBUF(35),NSAM)
34:       EQUIVALENCE (CATBUF(36),IDEL),(CATBUF(37),IDATE)
35:       EQUIVALENCE (CATBUF(40),IDAT1),(CATBUF(43),COMBUF)
36:       EQUIVALENCE (CATBUF(67),SHTLIN),(CATBUF(69),CCODE)
37:       LOGICAL SELFLG
38:       DATA CATLFN/30/,LFNS/40/,HDLFN/50/
39: C
40:       CALL BTIME
41:       OPEN CATLFN
42:       OPEN HDLFN
43: C POSITION TO FIRST EVENT RECORD IN CATALOG AREA; ALSO WRITE OUTPUT HEADLIN
44:         WRITE(LFNS,1000)
45: 1000  FORMAT(1X,' TAPE# INST# EVENT  FNUM     CHANNEL SELECT CODE',
46:      +'(1=SELECT, 0=DO NOT SELECT)')
47:       CALL DPOS(CATLFN,2)
48:       CALL BUFIN(CATLFN,CATBUF,112,IEOF)
49:       JHADR=CATBUF(112)
50:       FOR K=1,24
51:       .  CHSEL(K)=0
52:       END FOR
53:       NSEL=0
54:       NREC=0
55:       LOOP
56:       .  CALL BUFIN(CATLFN,CATBUF,112,IEOF)
57:       .  EXIT LOOP IF (IEOF.EQ.3)
58:       .  NREC=NREC+1
59:       .  EXIT LOOP IF(SIZE.GT.1.E+06)
60: C CHECK FOR HOW CHANNELS ENCODED
61:       .  IF(CCODE.EQ.1)
62:       .  .  FOR I=2,ICHN
63:       .  .  .  COMBUF(I)=COMBUF(1)
64:       .  .  END FOR
65:       .  END IF
66:       .  CALL RECSEL(ICSEL,SELFLG)
67:       .  IF(SELFLG)
68:       .  .  NSEL=NSEL+1
69:       .  .  IF(ICSEL(1).NE.9999)
70: C   SELECT CHANNELS REQUESTED
71:       .  .  .  FOR K=1,24
72:       .  .  .  .  FOR L=1,8
73:       .  .  .  .  .  IF(COMBUF(K).EQ.ICSEL(L))
74:       .  .  .  .  .  .  CHSEL(K)=1
75:       .  .  .  .  .  END IF
76:       .  .  .  .  END FOR
77:       .  .  .  END FOR
78:       .  .  END IF
79:       .  .  IF(ICSEL(1).EQ.9999)
80: C SELECT ALL CHANNELS
```

```
      . . . FOR K=1,ICHN
      . . . . CHSEL(K)=1
      . . . END FOR
      . . END IF
      . . WRITE(LFNS,1100) TAPENM,INUM,ENUM,FNUM,CHSEL
1100  . . FORMAT(416,4X,2412)
C FILL FILE HEADER BUFFER
      . . FHDR(1)=INUM
      . . FHDR(2)=TYPE
      . . FHDR(3)=ENUM
      . . CSEC=DST
      . . CALL CNTITM
      . . FOR I=1,7
      . . . FHDR(I+3)=TRAY(I)
      . . END FOR
      . . FHDR(11)=0
      . . FHDR(12)=IFIX(RANGE)
      . . FHDR(13)=IFIX(RANGE*1000.)-FHDR(12)*1000
      . . FHDR(14)=RERR
      . . FHDR(15)=IFIX(ILAT)
      . . FHDR(16)=IFIX(ILAT*1000.)-FHDR(15)*1000
      . . FHDR(17)=IFIX(ILON)
      . . FHDR(18)=IFIX(ILON*1000.)-FHDR(17)*1000
      . . FHDR(19)=IDEP
      . . FHDR(20)=WDEPI
      . . FHDR(21)=IFIX(ELAT)
      . . FHDR(22)=IFIX(ELAT*1000.)-FHDR(21)*1000
      . . FHDR(23)=IFIX(ELON)
      . . FHDR(24)=IFIX(ELON*1000.)-FHDR(23)*1000
      . . FHDR(25)=ELOER
      . . FHDR(26)=EDEP
      . . FHDR(27)=WDEPE
      . . CSEC=SBT
      . . CALL CNTITM
      . . FOR I=1,7
      . . . FHDR(I+27)=TRAY(I)
      . . END FOR
      . . FHDR(35)=EXPL
      . . IF(SIZE.GT.0.01) THEN
      . . . FHDR(36)=IFIX(ALOG10(SIZE*1000.)*1000.)
      . . ELSE
      . . . FHDR(36)=0
      . . . WRITE(3,222) ENUM,SIZE
222   . . . FORMAT(' EVENT SIZE LT OR = 0.01, EVENT: ',I6,F12.6)
      . . END IF
      . . IF(TYPE.EQ.1)
      . . . FHDR(36)=SIZE
      . . END IF
      . . FHDR(37)=BUB
      . . IF(SAMP.GT.999) SAMP=SAMP/100
      . . FHDR(38)=SAMP
      . . FHDR(39)=ICHN
      . . FHDR(40)=NWDS
      . . FHDR(41)=FNUM
      . . FHDR(60)=CCODE
      . . FOR K=1,ICHN
      . . . FHDR(41+K*20)=COMBUF(K)
      . . END FOR
      . . FHDR(71)=NRC
      . . FHDR(72)=NSAM
      . . BUFFER OUT(HDLFN,FHDR,B,256,MS,ML)
      . . CALL STATUS(HDLFN)
      . END IF
C ZERO CHANNEL SELECT ARRAY
      . FOR I=1,24
      . . CHSEL(I)=0
      . END FOR
      END LOOP
      CALL DSTAT(CATLFN,ISTAT,JRA)
      JRA=JRA-1
      CALL ETIME
      ENDFILE LFNS
      WRITE(LFNS,1200) NSEL,NREC,JHADR,JRA
1200  FORMAT(' NUMBER OF RECORDS SELECTED: ',I7,' TOTAL NUMBER OF',
     +' RECORDS PROCESSED: ',I7,/' HEADER ADDRESS:',I7,
     +' STOP ADDRESS:',I7)
      STOP
      END
```

```
 1: C
 2:       SUBROUTINE RECSEL(ICHN,SELFLG)
 3:       INTEGER ICHN(8),JBUF(112)
128       INTEGER TYPE,ENUM,EDEP,WDEPE,EXPLOS,WDEPI,TAPENM
 5:       INTEGER*6 STIME
 6:       REAL ILAT,ILON
 7:       COMMON /SEL/ JBUF
 8:       EQUIVALENCE (JBUF(2),TAPENM),(JBUF(3),INUM),(JBUF(4),ENUM)
 9:       EQUIVALENCE (JBUF(5),STIME),(JBUF(9),SIZE),(JBUF(11),RANGE)
10:       EQUIVALENCE (JBUF(13),ILAT),(JBUF(15),ILON),(JBUF(17),ELAT)
11:       EQUIVALENCE (JBUF(19),ELON),(JBUF(21),EXPLOS),(JBUF(22),WDEPI)
12:       EQUIVALENCE (JBUF(23),WDEPE),(JBUF(24),IDEP),(JBUF(25),EDEP)
13:       EQUIVALENCE (JBUF(27),TYPE),(JBUF(67),SHOTLN)
14:       LOGICAL SELFLG,QFLAG
15:       FOR I=1,8
16:       .  ICHN(I)=9999
17:       END FOR
18:         LOOP(1)
19:       .   QFLAG=.TRUE.
20:       .   LOOP(1)
21:       .   .   EXIT LOOP IF (INUM  .EQ.526)
22:       .   .   QFLAG = .FALSE.
23:       .   END LOOP
24:       .   EXIT LOOP IF (.NOT.QFLAG)
25:       .   LOOP(1)
26:       .   .   EXIT LOOP IF (SHOTLN.EQ."SLN3N ")
27:       .   .   QFLAG = .FALSE.
28:       .   END LOOP
29:       .   EXIT LOOP IF (.NOT.QFLAG)
30:        END LOOP
31:         IF (QFLAG)
32:       .    SELFLG=.TRUE.
33:         ELSE
34:       .    SELFLG=.FALSE.
35:         END IF
36:         RETURN
37:         END
```

```
 1: C ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 2: C PASCAL PROGRAM SELECT     BY MIKE SIMPSON
 3: C ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 4:      CONST
 5:      MAXNUMSTR = 15;
 6:      MAXRESVWDS = 20;
 7:      TYPE
 8:      TEXT = FILE OF CHAR;
 9:      CHTP = (LETTER,DIGIT,SPECIAL,ILLEGAL,SIGN,RELATION);
10:      NUMCASES = (NUMREAL,NUMINT,NUMDBL,NUMTIME,NUMALPHA);
11:      RESWDTYPES = (VALID,INVALID,USED,PAU,CHANNEL);
12:      ALPHA = ARRAY[1..6] OF CHAR;
13:      VAR
14:      DIAG : TEXT;
15:      CHARTP : ARRAY[CHAR] OF CHTP;
16:      ORDINT : ARRAY[CHAR] OF INTEGER;
17:      SOURCE : ARRAY[1..80] OF CHAR;
18:      INPLEN : INTEGER;
19:      CH : CHAR;
20:      EOL : BOOLEAN ;
21:      TEST : BOOLEAN;
22:      CHCNT : INTEGER;
23:      SCANWIDTH : INTEGER;
24:      PARAMSTRING : ALPHA;
25:      NUMSTRING : ARRAY[1..15] OF CHAR;
26:      NUMPTR : INTEGER;
27:      NUMTYPE : NUMCASES;
28:      RELSTRING : ARRAY[1..2] OF CHAR;  (* RELATION STRING *)
29:      RELNOT : BOOLEAN;  (* FLAG ON NOT RELATION *)
30:      RWTYPE : RESWDTYPES;
31:      RWSET : ARRAY[1..MAXRESVWDS] OF ALPHA;
32:      RWARG : ARRAY[1..MAXRESVWDS] OF RESWDTYPES;
33:      RWNUMTP : ARRAY[1..MAXRESVWDS] OF NUMCASES;
34:   PROCEDURE NEWLINE;
35:   BEGIN
36:     IF EOF(INPUT) THEN BEGIN
37:      WRITELN(OUTPUT,' **** EOF ****');
38:      SOURCE[1] := 'E';  SOURCE[2] := 'N';
39:      SOURCE[3] := 'D';  SOURCE[4] := '.';
40:      INPLEN := 4;
41:      TEST := FALSE;
42:      END
43:    ELSE BEGIN
44:      INPLEN := 0;
45:      REPEAT  READ(INPUT,CH);
46:              IF INPLEN < 80 THEN BEGIN
47:                INPLEN := INPLEN + 1;
48:                 SOURCE[INPLEN] := CH;
49:                 END;
50:        UNTIL EOLN(INPUT);
51:      READ(INPUT,CH);  (* READ BLANK AT EOL *)
52:          END;
53:          SCANWIDTH := INPLEN;
54:      CHCNT := 0;
55:      EOL := FALSE
56:  END (* NEWLINE *);
57:    PROCEDURE NEXTCH;
58:    BEGIN
59:      IF EOL THEN BEGIN
60:        NEWLINE;
61:        END;
62:      CHCNT := CHCNT + 1;
63:      EOL := CHCNT > SCANWIDTH;
64:      IF EOL THEN CH := ' ' ELSE CH := SOURCE[CHCNT];
65:    END (* NEXTCH *);
66:      PROCEDURE GETINTEGER;
67:      BEGIN
68:       WHILE CHARTP[CH] IN [DIGIT] DO
69:       BEGIN
70:       NUMSTRING[NUMPTR] := CH;
71:       NUMPTR := NUMPTR + 1;
72:       NEXTCH
73:       END
74:       END; (* GETINTEGER *)
75:       PROCEDURE NEXTIME;
76:       TYPE
77:       BETA = ARRAY[1..10] OF CHAR;
78:       VAR
79:       I,YRACC,LOCALACC : INTEGER;
80:       WORKSTR : BETA;
```

```
 81:       PROCEDURE BLDNUMERIC(VAR NUMACC : INTEGER);
 82:       BEGIN
 83:       NUMACC :=0;
 84:       WHILE CHARTP[CH] IN [DIGIT] DO BEGIN
 85:       NUMACC := NUMACC*10;
 86:       NUMACC := NUMACC + ORDINT[CH];
 87:       NEXTCH
 88:       END
 89:       END ;  (* BLDNUMERIC *)
 90:       PROCEDURE ADD(A,B : BETA; VAR C : BETA);
 91:       VAR
 92:       J,SUM,CARRY,DIGIT : INTEGER;
 93:       BEGIN
 94:       CARRY := 0;
 95:       FOR J := 10 DOWNTO 1 DO BEGIN
 96:       SUM := ORDINT[A[J]] + ORDINT[B[J]];
 97:       SUM := SUM + CARRY;
 98:       DIGIT := SUM MOD 10;
 99:       C[J] :=CHR(DIGIT + ORD('0'));
100:       CARRY := SUM DIV 10;
101:       END
102:       END;   (* ADD *)
103:       PROCEDURE MULT(MP1 :INTEGER; MP2 : BETA; VAR MP3 : BETA);
104:       VAR
105:       I,MDIGIT : INTEGER;
106:       BEGIN
107:       MP3 := '0000000000';
108:       WHILE MP1 > 0 DO BEGIN
109:       MDIGIT := MP1 MOD 10;
110:       FOR I := 1 TO MDIGIT DO ADD(MP2,MP3,MP3);
111:       FOR I := 1 TO 9 DO MP2[I] := MP2[I+1];
112:       MP2[10] := '0';
113:       MP1 := MP1 DIV 10
114:       END
115:       END;  (*  MULT  *)
116:       PROCEDURE NUMTOSTR(NTS1 : INTEGER; VAR NTS2 : BETA);
117:       VAR
118:       I,DIGIT : INTEGER;
119:       BEGIN
120:       FOR I := 10 DOWNTO 1 DO BEGIN
121:       DIGIT := NTS1 MOD 10;
122:       NTS2[I] := CHR(DIGIT + ORD('0'));
123:       NTS1 := NTS1 DIV 10
124:       END
125:       END ;  (*  NUMTOSTR  *)
126:       PROCEDURE ADDACC(MULPY : INTEGER);
127:       BEGIN
128:       IF CH = ',' THEN BEGIN
129:       WHILE NOT(CHARTP[CH] IN [DIGIT]) DO NEXTCH;
130:       BLDNUMERIC(LOCALACC);
131:       NUMTOSTR(LOCALACC,WORKSTR);
132:       MULT(MULPY,WORKSTR,WORKSTR);
133:       ADD(WORKSTR,NUMSTRING,NUMSTRING)
134:       END
135:       END;  (* ADDAC *)
136:       BEGIN
137:       BLDNUMERIC(YRACC);
138:       IF YRACC > 1900 THEN YRACC := YRACC - 1900;
139:       LOCALACC := YRACC DIV 4;
140:       YRACC := YRACC*365 + LOCALACC;
141:       NUMTOSTR(YRACC,NUMSTRING);
142:       MULT(86400,NUMSTRING,NUMSTRING);
143:       ADDACC(86400);
144:       ADDACC(3600);
145:       ADDACC(60);
146:       ADDACC(1);
147:       NUMPTR := 11;
148:       NUMSTRING[NUMPTR + 2] := '0';
149:       NUMSTRING[NUMPTR + 1] := '0';
150:       NUMSTRING[NUMPTR]     := '0';
151:       IF CH = ',' THEN
152:       NEXTCH;
153:       WHILE (CHARTP[CH] IN [DIGIT]) DO BEGIN
154:       NUMSTRING[NUMPTR] := NUMSTRING[NUMPTR + 1];
155:       NUMSTRING[NUMPTR + 1] := NUMSTRING[NUMPTR + 2];
156:       NUMSTRING[NUMPTR + 2] := CH;
157:       NEXTCH END;
158:       NUMPTR := NUMPTR + 4; NUMSTRING[NUMPTR - 1] := 'D'
159:       END ;   (* NEXTIME *)
160:       PROCEDURE GETNUMBER;
```

```
161:      BEGIN
162:      NUMPTR := 1;
163:      CASE NUMTYPE OF
164:      NUMALPHA :
165:      BEGIN
166:      WHILE CHARTP[CH] IN [LETTER,DIGIT] DO
167:      BEGIN
168:      IF CHARTP[CH] IN [LETTER] THEN
169:      BEGIN
170:      NUMSTRING[NUMPTR] :=CH;NUMPTR :=NUMPTR + 1;
171:      NEXTCH END;
172:      IF CHARTP[CH] IN [DIGIT] THEN GETINTEGER END;
173:      IF (NUMPTR = 3) THEN
174:      BEGIN
175:      NUMSTRING[NUMPTR] := ' ';NUMPTR := NUMPTR + 1;
176:      END
177:      END;  (* ALPHANUMERIC CASE *)
178:      NUMINT :
179:      BEGIN
180:      IF CHARTP[CH] IN [SIGN] THEN BEGIN
181:      NUMSTRING[NUMPTR] := CH; NUMPTR := NUMPTR + 1;
182:      NEXTCH END;
183:      GETINTEGER
184:      END;  (* NUMERIC INTEGER CASE *)
185:      NUMREAL :
186:      BEGIN
187:      IF CHARTP[CH] IN [SIGN] THEN
188:      BEGIN
189:      NUMSTRING[NUMPTR] := CH; NUMPTR := NUMPTR + 1; NEXTCH END;
190:      GETINTEGER;
191:      IF (CH = '.') THEN BEGIN
192:      NUMSTRING[NUMPTR] := '.';NUMPTR := NUMPTR + 1;
193:      NEXTCH;
194:      GETINTEGER
195:      END
196:      ELSE BEGIN
197:      NUMSTRING[NUMPTR] := '.'; NUMPTR := NUMPTR + 1 END
198:      END;     (* END OF NUMREAL STATEMENT *)
199:      NUMTIME   :
200:      NEXTIME
201:      END      (*  END OF CASE *)
202:      END;  (* GETNUMBER *)
203:      PROCEDURE NEXTRELATION;
204:      TYPE
205:      RELSPECIFIED = (RELSPEC,RELNOTSPEC);
206:      VAR
207:      RELTYPE : RELSPECIFIED;
208:      BEGIN
209:      IF CHARTP[CH] IN [RELATION] THEN
210:      RELTYPE := RELSPEC ELSE RELTYPE := RELNOTSPEC;
211:      CASE RELTYPE OF
212:      RELNOTSPEC :
213:      BEGIN
214:      RELSTRING[1] := 'E'; RELSTRING[2] := 'Q'; RELNOT := FALSE END;
215:      RELSPEC :
216:      BEGIN
217:      IF CH = '
                  THEN BEGIN RELNOT := TRUE;
218:      RELSTRING[1] := 'N'; RELSTRING[2] := 'E'; NEXTCH END
219:      ELSE
220:      BEGIN RELNOT := FALSE; RELSTRING[1] := 'E'; RELSTRING[2] := 'Q'
221:      END;
222:      IF CHARTP[CH] IN [RELATION] THEN
223:      CASE CH OF
224:      '<' :
225:      BEGIN
226:      IF RELNOT=FALSE THEN
227:      BEGIN RELSTRING[1] := 'L'; RELSTRING[2] := 'T'; NEXTCH END
228:      ELSE
229:      BEGIN RELSTRING[1] := 'G'; RELSTRING[2] := 'E'; NEXTCH END END;
230:      '>' :
231:      BEGIN
232:      IF RELNOT=FALSE THEN
233:      BEGIN RELSTRING[1] := 'G'; RELSTRING[2] := 'T'; NEXTCH END
234:      ELSE
235:      BEGIN RELSTRING[1] := 'L'; RELSTRING[2] := 'E'; NEXTCH END END;
236:      '=' :
237:      BEGIN RELSTRING[1] := 'E'; RELSTRING[2] := 'Q'; NEXTCH END;
238:      ' ' :
239:      END;  (* CASE < > = *)
240:      IF CH = '=' THEN BEGIN RELSTRING[2] := 'E'; NEXTCH END
```

```
241:      END (* RELSPEC CASE *)
242:      END (* RELTYPE CASE *)
243:      END; (* NEXTRELATION *)
244:      PROCEDURE EXPRESSION;
245:      TYPE
246:      EXPRANGE = (EXITYPE,HITYPE);
247:      VAR
248:      I : INTEGER;
249:      EXPTYPE : EXPRANGE;
250:      BEGIN
251:      WHILE NOT (CHARTP[CH] IN [LETTER,SIGN,DIGIT,RELATION])
252:      DO NEXTCH;
253:      IF CHARTP[CH] IN [SIGN,DIGIT,RELATION] THEN
254:      EXPTYPE := HITYPE ELSE EXPTYPE := EXITYPE;
255:      CASE EXPTYPE OF
256:      EXITYPE : ;
257:      HITYPE  :
258:      BEGIN
259:      NEXTRELATION;
260:      WRITE(OUTPUT,' ');
261:      WRITE(OUTPUT,'      EXIT LOOP IF (');
262:      GETNUMBER;
263:      IF CH = '/' THEN
264:      BEGIN
265:      WRITE(OUTPUT,'(');
266:      WRITE(OUTPUT,PARAMSTRING);
267:      WRITE(OUTPUT,'.GE.');
268:      FOR I := 1 TO (NUMPTR-1) DO WRITE(OUTPUT,NUMSTRING[I]);
269:      CASE NUMTYPE OF
270:      NUMINT,NUMREAL,NUMDBL :
271:      WRITE(OUTPUT,').AND.(');
272:      NUMTIME :
273:      BEGIN WRITE(OUTPUT,').AND.'); WRITELN(OUTPUT);
274:      WRITE(OUTPUT,'     +(') END
275:      END;
276:      WRITE(OUTPUT,PARAMSTRING);
277:      WRITE(OUTPUT,'.LE.');
278:      NEXTCH;
279:      GETNUMBER;
280:      FOR I := 1 TO (NUMPTR-1) DO WRITE(OUTPUT,NUMSTRING[I]);
281:      WRITE(OUTPUT,'))');
282:      WRITELN(OUTPUT);
283:      EXPRESSION
284:      END
285:      ELSE
286:      BEGIN
287:      WRITE (PARAMSTRING);
288:      WRITE(OUTPUT,'.',RELSTRING,'.');
289:      IF NUMTYPE = NUMALPHA THEN
290:      BEGIN
291:      WRITE(OUTPUT,'"SLN');
292:      FOR I :=1 TO NUMPTR - 1 DO WRITE(OUTPUT,NUMSTRING[I]);
293:      WRITE(OUTPUT,'"');
294:      END
295:      ELSE
296:      FOR I := 1 TO NUMPTR - 1 DO WRITE(OUTPUT,NUMSTRING[I]);
297:      WRITE(OUTPUT,')');
298:      WRITELN(OUTPUT);
299:      EXPRESSION
300:      END   (* IF / *)
301:      END   (* STATEMENT OF CASE *)
302:      END   (* CASE *)
303:      END;   (* EXPRESSION *)
304:      PROCEDURE TERM;
305:      VAR
306:      J,I : INTEGER;
307:      BEGIN
308:      I :=1;
309:      REPEAT
310:      IF I < 7 THEN BEGIN PARAMSTRING[I] :=CH; I := I+1 END;
311:      NEXTCH;
312:      UNTIL NOT(CHARTP[CH] IN [LETTER]);
313:      FOR J := I TO 6 DO PARAMSTRING[J] := ' ';
314:      RWTYPE := INVALID;
315:      FOR I := 1 TO MAXRESVWDS DO
316:      IF RWSET[I] = PARAMSTRING THEN BEGIN RWTYPE := RWARG[I];
317:      NUMTYPE := RWNUMTP[I];
318:      RWARG[I] := USED END;
319:      CASE RWTYPE OF
320:      INVALID :
```

```
321:      BEGIN
322:      WRITE(DIAG,' KEYWORD ',PARAMSTRING,' IS INVALID');
323:      WRITELN(DIAG)
324:      END;
325:      USED   :
326:      BEGIN
327:      WRITE(DIAG,' KEYWORD ',PARAMSTRING,' USED');
328:      WRITELN(DIAG);
329:      END;
330:      PAU     : ;
331:      CHANNEL : ;
332:      VALID   :
333:      BEGIN
334:      WRITE(OUTPUT,'       LOOP(1)');
335:      WRITELN(OUTPUT);
336:      EXPRESSION;
337:      WRITE(OUTPUT,'       QFLAG = .FALSE.');
338:      WRITELN(OUTPUT);
339:      WRITE(OUTPUT,'       END LOOP');
340:      WRITELN(OUTPUT);
341:      WRITE(OUTPUT,'       EXIT LOOP IF (.NOT.QFLAG)');
342:      WRITELN(OUTPUT)
343:      END;
344:      END
345:      END;   (*  TERM  *)
346:      PROCEDURE CHSET;
347:      VAR KK,I : INTEGER;
348:      BEGIN
349:      IF RWSET[17] = PARAMSTRING THEN
350:      BEGIN
351:      KK :=1;
352:      REPEAT
353:      NEXTCH;
354:      WHILE CHARTP[CH] IN [DIGIT] DO
355:      BEGIN
356:      GETNUMBER;
357:      WRITE(OUTPUT,'       ICHN(',KK,')=');
358:      FOR I :=1 TO (NUMPTR-1) DO WRITE(OUTPUT,NUMSTRING[I]);
359:      WRITELN;
360:      KK := KK+1;
361:      END
362:      UNTIL CHARTP[CH] IN [LETTER]
363:      END
364:      ELSE
365:      END;              (* CHSET *)
366:      PROCEDURE SETRWSET;
367:      BEGIN
368:      RWSET[1] := 'RANGE '; RWSET[2] := 'EXPLOS';
369:      RWSET[3] := 'TYPE  '; RWSET[4] := 'ETIME ';
370:      RWSET[5] := 'ENUM  '; RWSET[6] := 'STIME '; RWSET[7] := 'ELAT  ';
371:      RWSET[8] := 'ELON  '; RWSET[9] := 'EDEP  '; RWSET[10] := 'WDEPE ';
372:      RWSET[11] := 'SIZE  '; RWSET[12] := 'INUM  ';
373:      RWSET[13] := 'ILAT  '; RWSET[14] := 'ILON  ';
374:      RWSET[15] := 'IDEP  '; RWSET[16] := 'WDEPI ';
375:      RWSET[17] := 'CHAN  ';
376:      RWSET[18] := 'SHOTLN'; RWSET[19] := 'TAPENM';
377:      RWSET [20] := 'END   ';
378:      END;
379:      PROCEDURE SETRWARG;
380:      BEGIN
381:      RWARG[1] := VALID; RWARG[2] := VALID; RWARG[3] := VALID;
382:      RWARG[4] := VALID; RWARG[5] := VALID; RWARG[6] := VALID;
383:      RWARG[7] := VALID; RWARG[8] := VALID; RWARG[9] := VALID;
384:      RWARG[10] := VALID; RWARG[11] := VALID; RWARG[12] := VALID;
385:      RWARG[13] := VALID; RWARG[14] := VALID; RWARG[15] := VALID;
386:      RWARG[16] := VALID; RWARG[17] := CHANNEL;
387:      RWARG[18] := VALID; RWARG[19] := VALID;
388:      RWARG[20] := PAU;
389:      END;
390:      PROCEDURE SETRWNUM;
391:      BEGIN
392:      RWNUMTP[1] := NUMREAL; RWNUMTP[2] := NUMINT;
393:      RWNUMTP[3] := NUMINT;  RWNUMTP[4] := NUMTIME;
394:      RWNUMTP[5] := NUMINT;
395:      RWNUMTP[6] := NUMTIME; RWNUMTP[7] := NUMREAL;
396:      RWNUMTP[8] := NUMREAL; RWNUMTP[9] := NUMINT;
397:      RWNUMTP[10] := NUMINT; RWNUMTP[11] := NUMREAL;
398:      RWNUMTP[12] := NUMINT; RWNUMTP[13] := NUMREAL;
399:      RWNUMTP[14] := NUMREAL; RWNUMTP[15] := NUMINT;
400:      RWNUMTP[16] := NUMINT; RWNUMTP[17] := NUMINT;
```

```
401:      RWNUMTP[18] := NUMALPHA; RWNUMTP[19] := NUMINT;
402:      RWNUMTP[20] := NUMINT;
403:      END;
404:      PROCEDURE PROLOG;
405:      BEGIN
406:      WRITE(OUTPUT,'       LOOP(1)');
407:      WRITELN(OUTPUT);
408:      WRITE(OUTPUT,'       QFLAG=.TRUE.');
409:      WRITELN(OUTPUT)
410:      END;
411:      PROCEDURE JOB;
412:      BEGIN
413:      PROLOG;
414:      REPEAT
415:      WHILE NOT(CHARTP[CH] IN [LETTER]) DO NEXTCH;
416:      TERM;
417:      CHSET;
418:      UNTIL RWTYPE = PAU;
419:      WRITE(OUTPUT,'       END LOOP');
420:      WRITELN(OUTPUT);
421:      WRITE(OUTPUT,'        IF (QFLAG)');
422:      WRITELN(OUTPUT);
423:      WRITE(OUTPUT,'        SELFLG=.TRUE.');
424:      WRITELN(OUTPUT);
425:      WRITE(OUTPUT,'        ELSE');
426:      WRITELN(OUTPUT);
427:      WRITE(OUTPUT,'        SELFLG=.FALSE.');
428:      WRITELN(OUTPUT);
429:      WRITE(OUTPUT,'        END IF');
430:      WRITELN(OUTPUT);
431:      WRITE(OUTPUT,'        RETURN');
432:      WRITELN(OUTPUT);
433:      WRITE(OUTPUT,'        END');
434:      WRITELN(OUTPUT);
435:      END;
436:     PROCEDURE CHARTYPES;
437:     VAR I : INTEGER;
438:     BEGIN
439:       FOR I := 32 TO 95 DO CHARTP[CHR(I)] := ILLEGAL;
440:       CHARTP['A'] := LETTER ;
441:       CHARTP['B'] := LETTER ; CHARTP['C'] := LETTER ;
442:       CHARTP['D'] := LETTER ; CHARTP['E'] := LETTER ;
443:       CHARTP['F'] := LETTER ; CHARTP['G'] := LETTER ;
444:       CHARTP['H'] := LETTER ; CHARTP['I'] := LETTER ;
445:       CHARTP['J'] := LETTER ; CHARTP['K'] := LETTER ;
446:       CHARTP['L'] := LETTER ; CHARTP['M'] := LETTER ;
447:       CHARTP['N'] := LETTER ; CHARTP['O'] := LETTER ;
448:       CHARTP['P'] := LETTER ; CHARTP['Q'] := LETTER ;
449:       CHARTP['R'] := LETTER ; CHARTP['S'] := LETTER ;
450:       CHARTP['T'] := LETTER ; CHARTP['U'] := LETTER ;
451:       CHARTP['V'] := LETTER ; CHARTP['W'] := LETTER ;
452:       CHARTP['X'] := LETTER ; CHARTP['Y'] := LETTER ;
453:       CHARTP['Z'] := LETTER ; CHARTP['0'] := DIGIT ;
454:       CHARTP['1'] := DIGIT ; CHARTP['2'] := DIGIT ;
455:       CHARTP['3'] := DIGIT ; CHARTP['4'] := DIGIT ;
456:       CHARTP['5'] := DIGIT ; CHARTP['6'] := DIGIT ;
457:       CHARTP['7'] := DIGIT ; CHARTP['8'] := DIGIT ;
458:       CHARTP['9'] := DIGIT ; CHARTP['+'] := SIGN   ;
459:       CHARTP['-'] := SIGN   ; CHARTP['*'] := SPECIAL;
460:       CHARTP['/'] := SPECIAL; CHARTP['('] := SPECIAL;
461:       CHARTP[')'] := SPECIAL; CHARTP['$'] := SPECIAL;
462:       CHARTP['='] := RELATION; CHARTP[' '] := SPECIAL;
463:       CHARTP[','] := SPECIAL; CHARTP['.'] := SPECIAL;
464:       CHARTP[''''] := SPECIAL; CHARTP['['] := SPECIAL;
465:       CHARTP[']'] := SPECIAL; CHARTP[':'] := SPECIAL;
466:       CHARTP['"'] := SPECIAL; CHARTP[';'] := SPECIAL;
467:       CHARTP['<'] := RELATION; CHARTP['>'] := RELATION;
468:      CHARTP['^'] := RELATION;
469:       ORDINT['0'] := 0; ORDINT['1'] := 1; ORDINT['2'] := 2;
470:       ORDINT['3'] := 3;
471:       ORDINT['4'] := 4; ORDINT['5'] := 5; ORDINT['6'] := 6;
472:      ORDINT['7'] := 7; ORDINT['8'] := 8; ORDINT['9'] := 9;
473:     END;
474:      PROCEDURE HEADLD;
475:      FORTRAN;
476:      BEGIN
477:      HEADLD;
478:      RESET(DIAG);
479:      WRITELN(DIAG,' CPU TIME = ':15,CLOCK:6);
480:      CHARTYPES;
```

```
481:      SETRWSET;
482:      SETRWARG;
483:      SETRWNUM;
484:      WRITE(' ');
485:      NEWLINE;
486:      NEXTCB;
487:      JOB;
488:      WRITELN(DIAG,' CPU TIME = ':15,CLOCK:6)
489:      END.
```

```
 1:       SUBROUTINE HEADLD
 2:       INTEGER INAME1(3),INAME2(3),IBUF(27)
 3:       DATA INAME1/"RECSEL   "/,INAME2/"SUBOUT   "/
 4:       CALL ASSIGN(10,INAME1,IERR)
 5:       CALL ASSIGN(7,INAME2,IERR)
 6:       LOOP
 7:       BUFFER IN(10,IBUF,S,27,ISTAT,ILEN)
 8:       CALL STATUS(10)
 9:       EXIT LOOP IF(ISTAT.GE.3)
10:       BUFFER OUT(7,IBUF,S,27,ISTAT,ILEN)
11:       CALL STATUS(7)
12:       END LOOP
13:       RETURN
14:       END
```

```
 1:* FILENAME HEAD; HEADING LINES FOR SELECTION SUBROUTINE
 2:       SUBROUTINE RECSEL(ICHN,SELFLG)
 3:       INTEGER ICHN(8)
 4:       INTEGER TYPE,ENUM,EDEP,WDEPE,SIZE,EXPLOS,WDEPI,TAPENM
 5:       INTEGER*6 STIME
 6:       REAL ILAT,ILON
 7:       COMMON /KEYS/ TAPENM,INUM,ENUM,STIME,SIZE,RANGE,ILAT,ILON,
 8:      + ELAT,ELON,EXPLOS,WDEPI,WDEPE,IDEP,EDEP,TYPE,SHOTLN
 9:       LOGICAL SELFLG,QFLAG
10:       FOR I=1,8
11:       ICHN(I) = 9999
12:       END FOR
```

```
 1: $MS
 2: $RW 3
 3: $MO RE
 4: $PR MACRO M<SEARCH TO GENERATE SELECTION CRITERIA & SEARCH CATALOG
 5: $AS 10=T1
 6: $PR ARE INPUT PARAMETERS IN FILE PASCIN? (1)
 7: $PR OR IS INPUT FROM THE TERMINAL?       (2)
 8: $SR.IN #N
 9: IF (#N=2) $JU !TER
10: $AS 20=PASCIN
11: $JU !CON
12: !TER AS 20=*0
13: !CON $$XSELECT
14: $FR 5 6
15: $FO.P RECSEL
16: $PR SUBROUTINE COMPILED
17: FR 6
18: $CO LO S2
19: $JS SRCHVU
20: $PR MAIN AND SUB VULCANIZED
21: $PR WANT TO CONTINUE AND SEARCH CATALOG? ENTER Y OR TO STOP ENTER N
22: $SR.IT #ANS
23: IF (#ANS="N") $JU !END
24: PR RUN FROM TERMINAL OR CONTROL POINT (T OR C)?
25: SR.IT #ANS
26: IF (#ANS="T") $JU !TERM
27: LJ JSRCH
28: PR JOB JSRCH INSERTED. LIST S2 FOR RECSEL, LISTOUT FOR JOB
29: PR AND SCOUT FOR DATA SELECTED
30: $ME
31: !TERM
32: AS 6=*0
33: AS 30=REVCAT.B
34: AS 40=T2
35: AS 50=SRCHHD
36: PR RUNNING CATALOG SEARCH NOW
37: $$XSEARCH
38: !END PR M<SEARCH ALL PAU: LIST S2 FOR RECSEL, T2 FOR SELECTION CRITERIA
39: $ME
```

```
 1: * PROGRAM RETREV4 TO RETRIEVE ARCHIVED EVENTS AS SPECIFIED BY A
 2: * PARTICIPANT'S REQUEST.  USES AS INPUT THE OUTPUT FROM XSEARCH(SEARCH:3)
 3: * (2 FILES, ONE WITH RECORDS TO SELECT, THE OTHER WITH EVENT HEADERS)
 4: * AND A TAPE.  OUTPUTS ARE (1) A DATA FILE OF SELECTED EVENTS AND
 5: * (2) A TAPE HEADER FILE IN WHICH THE NUMBER OF EVENTS REFLECTS
 6: * ALSO REMOVES MEAN FROM UW DATA
 7: *
 8: *                   USE MACRO M<RETREV    COMPILE WITH J.RET
 9: *
10: *                         SAUF77.I RETREV4
11: *                         VU.R XRETREV4 PA=3
12: *                         LIB 1500MGG*MRSLIB *SAUL77 *LIBERY
13: *                         AS 88=T1
14: *
15: *      LAST MODIFIED 3/08/83       SLL
16: *
17:       INTEGER THBUF(256),KBUF(256),IBUF(8192),TAPE,SELIN,OUTF,HDOUT
18:       INTEGER AFILES(4),ENUMS,FNUM,LBUF(256),FHLFN
19:       INTEGER ENUMMX,ENUMMN,TRAY(7),TRAY1(7)
20:       INTEGER*6 TDST,TDET
21:       DIMENSION DNAME(9),EXCODE(4),SHTLN(10),XF(4096)
22:       EQUIVALENCE (XF,IBUF)
23:       COMMON/ITCM/ TRAY,CSEC,JULD
24:       DATA TAPE/4/,SELIN/15/,HDOUT/11/,OUTF/20/,FHLFN/50/
25:       OPEN TAPE
26:       OPEN FHLFN
27:       ICNT=0
28:       ISTOP=0
29:       READ(SELIN,100,END=999) JTAP,INUMS,ENUMS,FNUM
30: 100   FORMAT(4I6)
31: * Save tape number and IID number just read
32:       LTAP=JTAP
33:       LINUM=INUMS
34: * Find HIG tape header file - should be first file of each Archive
35:       DO
36:       .  BUFFER IN (TAPE,IBUF,B,4096,IS,IL)
37:       .  CALL STATUS(TAPE)
38:       UNTIL(IL.EQ.224)
39:       DECODE(9,4000,IBUF) ITAP
40: 4000  FORMAT(3X,I6)
41:       IF(ITAP.NE.JTAP) STOP TAPENM
42:       DECODE(174,6002,IBUF(95)) IID,DNAME,EXCODE,NFILES,TDST,TDET,SHTLN
43: 6002  FORMAT(I4,12A6,A2,I6,I14,1X,I14,1X,10A6)
44:       WRITE(3,2001) ITAP
45: 2001  FORMAT(/1X,'ROSE ARCHIVE TAPE NO.',T40,I6)
46:       WRITE(3,2027)
47: 2027  FORMAT(1X,'***** TAPE HEADER FILE CONTENTS *****')
48:       WRITE(3,2028) IID,DNAME
49: 2028  FORMAT(1X,'INSTR. #',T20,I4,'    DESIGNER',T40,9A6)
50:       WRITE(3,2029) EXCODE,NFILES
51: 2029  FORMAT(1X,'EXPERIMENT:',T20,3A6,A2,'   # OF EVENTS:',
52:      +T60,I6)
53: * Save ROSE Archive tape header
54:       DO
55:       .  BUFFER IN (TAPE,THBUF,B,256,IS,IL)
56:       .  CALL STATUS(TAPE)
57:       UNTIL (IL.EQ.256)
58:       FOR II=1,256
59:       .  KBUF(II)=THBUF(II)
60:       END FOR
61: * Get data from tape
62: 1     LOOP                    !for 1 event
63:       .  DO
64:       .  .  BUFFER IN (TAPE,IBUF,B,4096,IS,IL)
65:       .  .  CALL STATUS(TAPE)
66:       .  .  IF(IS.GT.3)
67:       .  .  .  WRITE(3,155)
68: 155   .  .  .  FORMAT(' EOF ON INPUT')
69:       .  .  .  GO TO 999
70:       .  .  END IF
71:       .  UNTIL(IL.EQ.256)
72:       .  IF(IBUF(1).NE.INUMS)
73:       .  .  WRITE(3,"(' INSTRUMENT # ON TAPE: ',I5,'   DOES NOT MATCH',
74:      +.  .  ' INSTRUMENT # OF SEARCH: ',I5)") IBUF(1),INUMS
75:       .  .  STOP INUM
76:       .  END IF
77:       .  IF(ENUMS.NE.0)
78: * Look for event or file # match
79:       .  .  EXIT LOOP IF(ENUMS.EQ.IBUF(3))
80:       .  .  EXIT LOOP IF(FNUM.EQ.IBUF(41))
```

```
81:       . END IF
82:       . IF(ENUMS.EQ.0)
83:       . . EXIT LOOP IF(FNUM.EQ.IBUF(41))
84:       . END IF
85:       END LOOP
86: * Read event headers from search file (which comes from Revcat)
87: 2     BUFFER IN(FHLFN,LBUF,B,256,IS,IL)
88:       CALL STATUS(FHLFN)
89:       IF(IS.GE.3)
90:       . WRITE(3,2050) ENUMS,INUMS
91: 2050  . FORMAT(' Event ',I6,' or Rcv ',I6,' not found in header file',
92:      +. /' rewinding header file')
93:       . ISTOP=ISTOP+1
94:       . IF(ISTOP.GE.5) STOP ERROR
95:       . REWIND 50
96:       . GOTO 2
97:       END IF
98: * Match event# and inst#
99:       IF(LBUF(3).NE.ENUMS) GO TO 2
100:      IF(LBUF(1).NE.INUMS) GO TO 2
101:      NC=LBUF(39)
102:      ISK=41
103:      LLR=LBUF(72)
104: * Fill in words 62-70
105:      FOR J=1,NC
106:      . FOR I=42,50
107:      . . LBUF(I+20*J)=IBUF(I+20*J)
108:      . END FOR
109:      END FOR
110:      IF(LLR.EQ.0)
111:      . IF(ICNT.EQ.0)
112:      . . WRITE(3,"('Using tape headers instead of those from REVCAT')")
113:      . END IF
114:      . FOR J=1,NC
115:      . . FOR I=ISK,52
116:      . . . LBUF(I+20*J)=IBUF(I+20*J)
117:      . . END FOR
118:      . END FOR
119:      . LLR=IBUF(72)
120:      END IF
121:      ICNT=ICNT+1
122:      LBUF(41)=ICNT
123:      IF(ICNT.EQ.1)
124:      . ENUMMN=ENUMS
125:      . FOR I=1,7
126:      . . TRAY(I)=LBUF(I+3)
127:      . END FOR
128:      END IF
129:      BUFFER OUT(OUTF,LBUF,B,256,IS,IL)
130:      CALL STATUS(OUTF)
131:      IF(IS.GT.3)
132:      . WRITE(3,107)
133: 107  . FORMAT(' EOT ON OUTPUT')
134:      . STOP ERROR
135:      END IF
136:      WRITE(3,110) LBUF(3),LBUF(1),FNUM,ICNT
137: 110  FORMAT(' RETRIEVING EVENT #',I6,'  RCV  ',I4,
138:     +' OLD F# ',I4,'  NEW F# ',I4)
139:      K=LBUF(71)
140:      N=(LBUF(39))*K
141:      JCNT=0
142: * Loop to get data
143:      LOOP
144:      . BUFFER IN (TAPE,IBUF,B,4096,IS,IL)
145:      . CALL STATUS(TAPE)
146:      . LTH=4096
147:      . IF(IS.GE.3)
148:      . . IF(JCNT.LT.N) WRITE(10,"(' RECORDS READ LESS THAN SPECIFIED',
149:      +. . ' IN HEADER',3I6)") ENUMS, N, JCNT
150:      . . IF(JCNT.GT.N) WRITE(10,"(' RECORDS READ MORE THAN SPECIFIED',
151:      +. . ' IN HEADER',3I6)") ENUMS, N, JCNT
152:      . . EXIT LOOP
153:      . END IF
154:      . JCNT=JCNT+1
155:      . IF(IID.GE.400.AND.IID.LE.499)
156: * Convert integer to real thru equiv arrays and remove mean
157:      . . FOR I=4096,1,-1
158:      . . . XF(I)=IBUF(I)
159:      . . END FOR
160:      . . IF(JCNT.EQ.K)
```

```
161:        . . .  LTH=LLR
162:        . . .  JCNT=0
163:        . .  END IF
164:        . .  CALL MEAN(XF,LTH)
165:        . .  FOR I=1,LTH
166:        . . .  IBUF(I)=XF(I)
167:        . .  END FOR
168:        . .  IF(LTH.LT.4096)
169:        . . .  L=LLR+1
170:        . . .  FOR I=L,4096
171:        . . . .  IBUF(I)=0
172:        . . .  END FOR
173:        . .  END IF
174:        .  END IF
175:        .  BUFFER OUT(OUTF,IBUF,B,4096,IS,IL)
176:        .  CALL STATUS(OUTF)
177:        .  IF(IS.GT.3)
178:        . .  WRITE(3,107)
179:        . .  STOP ERROR
180:        .  END IF
181:        .  ENUMMX=ENUMS
182:       END LOOP
183:       ENDFILE OUTF
184:       READ(SELIN,100,END=999) JTAP,INUMS,ENUMS,FNUM
185:       IF(JTAP.NE.LTAP) GO TO 888
186:       IF(INUMS.NE.LINUM) GOTO 888
187: C      IF(INUMS.NE.IID)
188: C      IF(ICNT.EQ.1)
189: C      WRITE(3,"(' IID ON TAPE=',I4,'  INUMS=',I4)")IID,INUMS
190: C      END IF
191: C      END IF
192:       GO TO 1
193: *  Encode ICNT into NFILES in header
194: 888    BACKSPACE SELIN
195:       WRITE(3,166)
196: 166    FORMAT(' NORMAL EXIT')
197: 999    WRITE(3,120) ICNT
198: 120    FORMAT(' NUMBER OF FILES=',I6)
199:       FOR I=1,7
200:       .  TRAY1(I)=LBUF(I+3)
201:       END FOR
202:       WRITE(88,140) ICNT
203: 140    FORMAT(I8)
204:       BACKSPACE 88
205:       READ(88,130) AFILES
206: 130    FORMAT(4R2)
207:       FOR I=1,4
208:       .  KBUF(50+I)=AFILES(I)
209:       END FOR
210:       BUFFER OUT(HDOUT,KBUF,B,256,IS,IL)
211:       WRITE(10,2001) LTAP
212:       WRITE(10,2027)
213:       WRITE(10,2028) LINUM,DNAME
214:       WRITE(10,2035) EXCODE,ICNT
215: 2035   FORMAT(' EXPERIMENT: ',T20,3A6,A2,'  # OF EVENTS RETRIEVED:',
216:      +T70,I6)
217:       WRITE(10,"(' EVENT #S: ',I6,' - ',I6)") ENUMMN,ENUMMX
218:       WRITE(10,"(' START TIME OF FIRST EVENT:',7I4)") TRAY
219:       WRITE(10,"(' START TIME OF LAST EVENT:',7I4)") TRAY1
220:       STOP
221:       END

222:       SUBROUTINE MEAN(XF,NPT)
223:       DIMENSION XF(1)
224:       XM=XF(1)
225:       DO 1 I=2,NPT
226:    1  .  XM=XM+XF(I)
227:       XM=XM/NPT
228:       DO 2 I=1,NPT
229:    2  .  XF(I)=XF(I)-XM
230:       RETURN
231:       END
```

```
 1: $MS
 2: $PR Macro "1 5 1 2 R O S E * M < R E T R E V"
 3: $PR Macro to retrieve archived ROSE data,
140
 4: PR and write it to tape.                    Last modified by CM  2/16/82.
 5: PR
 6: IF,(.NOT.(C.SPA.A&0)) $JU !DRI
 7: $PR Enter name of header file. "i.e. TEH.@@"
 8: $SR.IT #FIL
 9: $AS 11=#FIL
10: $PR Enter name of data file.  "i.e. TED.@@"
11: $SR.IT #FIL
12: $AS 20=#FIL
13: PR Tape-drive 12 is:
14: /PS 12
15: PR Tape-drive 10 is:
16: /PS 10
17: PR Tape-drive 9 is:
18: /PS 9
19: PR Enter drive No. to resource.
20: $SR.IN #NUM
21: $JU !COPY
22: !DRI PR Tape-drive 12 is:
23: /PS 12
24: PR Tape-drive 10 is:
25: /PS 10
26: PR Tape-drive 9 is:
27: /PS 9
28: PR Enter drive no. to resource.
29: $SR.IN #NUM
30: PR Mount archive tape on drive no. #NUM
31: !MOUNT RS 4=ARCTAP 1600B  WA :#NUM
32: !TEM PR Do you want to assign file names? Yes / blank card for not.
33: SR.IT #IYS
34: IF (#IYS) JU !GEN
35: PR Do you want to stop? Yes / blank card for not.
36: SR.IT #IYS
37: IF (#IYS) JU !END
38: JU !SKIP
39: !GEN RW 4
40: PR Enter name for data output file: "TED.@@".
41: SR.IT #FIL
42: $GE #FIL G500 M170000 BF=7 OW PD PR P3
43: JE 2150 !ASD
44: !ASD AS 20=#FIL
45: PR Enter name of header file: "TEH.@@".
46: SR.IT #HED
47: GE #HED G=2 OW PD PR P3
48: JE 2150 !ASH
49: !ASH AS 11=#HED
50: PR Enter name for report file: "TER.@@".
51: SR.IT #REP
52: GE #REP G=10 OW PR PD P3
53: JE 2150 !ASR
54: !ASR AS 10=#REP
55: PR Enter the name of the SCOUT file: "SCOUT.@@".
56: SR.IT #SCO
57: AS 15=#SCO
58: AR 15
59: PR Enter the name of the search head file: "SRCHD.@@"
60: SR.IT #SRC
61: AS 50=#SRC
62: !SKIP PR Retrieval starting from first (or this) file? Yes / blank
63: SR.IT #YES
64: IF (#YES)  JU !EXCUT
65: PR Do you want to retrieve manually? Yes (get out of the macro)/ blank
66: IF (#YES)  JU !MANUA
67: $$XTAPOS
68: !EXCUT $1512ROSE*XRETREV4
69: !MAC MA.E #FIL
70: JE.P 37 !MAG
71: PR More to retrieve from this tape? Yes / blank card for not.
72: SR.IT #YES
73: IF,(#YES) $JU !TEM
74: PR More to retrieve from another tape? Yes / blank card for not.
75: SR.IT #YES
76: IF,(#YES) $JU !MORE
77: JU !FREE
78: !MORE RW 4
79: PR Mount next tape.
80: JU !TEM
```

```
 81: !FREE FR 4
 82: RW 11
 83: RW 20
 84: AS 12=T1
 85: WI 12
 86: CO #REP 12
 87: PR Dismount archive tape.
 88: PR
 89: PR Copy disc files of retrieved data to tape? Yes / blank card for not.
 90: SR.IT #YES
 91: IF,(#YES) JU !COPY
 92: JU !END
 93: !COPY PR Enter density 800 or 1600.
 94: SR.IN #DEN
 95: PR Load tape for retrieved data on drive no. #NUM
 96: PR
 97: SR.N #NOM=1
 98: IF,(#DEN=800) JU !EIGHT
 99: RS 4=TRANSMIT 2C  1600B WR WA :#NUM
100: JU !WRIT
101: !EIGHT RS 4=TRANSMIT 2C 800B WR WA :#NUM
102: !WRIT PR Skip data set?  Yes / blank card for not.
103: SR.IT #YES
104: IF (#YES) JU !XSK
105: JU !WROT
106: !XSK XSKIP
107: !WROT $$XWRITP
108: PR Data for data set #NOM written to transmittal tape.
109: SR.N #NOM=#NOM+1
110: PR More data in this file?  Yes / blank card for not.
111: SR.IT #YES
112: IF (#YES) JU !WRIT
113: PR Write data from another file? Yes / blank card for not.
114: SR.IT #YES
115: SR.N #NOM=1
116: IF,(#YES) $JU !EDF
117: JU !CHK
118: !EDF PR Enter data file name.
119: SR.IT #FIL
120: AS 20=#FIL
121: PR Enter header file name.
122: SR.IT #FIL
123: AS 11=#FIL
124: JU !WRIT
125: !CHK PR Now do check read.
126: RW 4
127: AS 10=T3
128: XDISTAP
129: PR List T3 for contents of tape header.
130: FR 4
131: PR Retrieval Pau - Dismount tape, log and send.
132: PR Eliminate data out area.
133: !END  PR Next: To join data and head file, and demultiplex data use
134: PR   1512REF*JCOM.TEX, JCOM.WHO, JCOM.UW or JCOM.ARC. (must be edited).
135: PR A L L  P A U U . . . U,              Se acabo esto!
136: ALL
137: $ME
138: !MAG FATAL MAG TAPE ERROR
139: $ME
140: !MANUA PR Next:  advance files   (AF 4 num.-files)
141: PR         print    JS M<RETREV !EXCUT    for going back to the macro.
142: $ME
```

```
 1:       NAME RARHIG
 2: C  PROGRAM TO CONVERT ROSE ARCHIVED DATA TO HIG DEMUX DATA
 3: C USE DATA FILE OUTPUT OF ARCHIVE AS INPUT FILE; OR LOAD DATA FROM
 4: C THE ARCHIVE TAPE AND USE THAT.  WHEN YOU LOAD FROM TAPE, BE SURE TO
 5: C ADVANCE ONE FILE BEFORE YOU COPY DATA TO DISC OR IN JOBSTREAM.
 6: C
 7: C       LAST UPDATED 5/13/83
 8: C            AS 20=ROSE FORMATTED INPUT DATA FILE
 9: C            AS 40=CORFILE
10: C            AS 10=T1 (TEMP FILE FOR HEADERS)
11: C            AS 61-N=OUTPUT DEMUX DATA FILES (1 PER CHANNEL)
12:       INTEGER MBUF(4032),IRECNO(500,24),INEXT(24)
13:       INTEGER*1 ICRAY(162),DCBUF(162)
14:       INTEGER IBUF(4096),DBUF(54),CATBUF(112),THBUF(224)
15:       INTEGER WDSLFT,CORFIL,HDFIL,ENUMMN,ENUMMX
16:       INTEGER ENUM,EXPL,WDEPI,WDEPE,EDEP,TYPE,RERR,ELOER
17:       INTEGER BUB,SAMP,FNUM,COMBUF(24),CCODE,CRA
18:       REAL ILAT,ILON
19:       INTEGER*6 CSEC,STATS,TDST,DST,SBT
20:       DIMENSION DNAME(9),EXCODE(4),A(4)
21:       LOGICAL HDRSW
22:       INTEGER FHDR(260),CORBUF(140),TRAY(7),HDREC(17)
23:       COMMON /ITCM/TRAY,CSEC,JULD
24:       EQUIVALENCE (IBUF(1),ICRAY(1)),(DBUF,DCBUF)
25:       DATA INFILE/20/,CORFIL/40/,HDFIL/10/
26:       DATA HDRSW/.TRUE./
27:       BUFFER IN(20,IBUF,B,4096,ISTAT,ILEN)
28:       CALL STATUS(20)
29:       JUMP=0
30:       J=1
31:       FOR I=1,162
32:       .  K=MOD(I,3)
33:       .  IF (K.NE.1)
34:       .  .  DCBUF(J)=ICRAY(I)
35:       .  .  J=J+1
36:       .  END IF
37:       END FOR
38: C DECODE NUMBER OF FILES
39:       DECODE(9,"(1X,BN,I8)",DBUF(34)) IFILES
40:       DECODE(4,"(BN,I4)",DBUF)IID
41:       DECODE(55,8005,DBUF(2)) DNAME
42: 8005  FORMAT(1X,9A6)
43:       DECODE(20,8006,DBUF(21)) EXCODE
44: 8006  FORMAT(3A6,A2)
45:       DECODE(12,8010,DBUF(27))ISYR,ISMO,ISDA,ISHR,ISMIN
46:  8010 FORMAT(2X,5I2)
47:       DECODE(10,8020,DBUF(31))IFYR,IFMO,IFDA,IFHR,IFMIN
48:  8020 FORMAT (5I2)
49:       WRITE(3,8040)IID,IFILES
50:  8040 FORMAT(1X,"INSTRUMENT ID  ",I4,/," # FILES  ",I4)
51:       WRITE(3,8045) DNAME
52: 8045  FORMAT(1X,'DESIGNER NAME & ADDRESS: ',9A6)
53:       WRITE(3,8046) EXCODE
54: 8046  FORMAT(1X,3A6,A2,' EXPERIMENT')
55:       WRITE(3,8050)ISYR,ISMO,ISDA,ISHR,ISMIN
56: 8050  FORMAT(1X,"START TIME  ",5(1X,I2))
57:       WRITE(3,8060)IFYR,IFMO,IFDA,IFHR,IFMIN
58: 8060  FORMAT(1X,"END TIME    ",5(1X,I2))
59:       FOR I=1,140
60:       .  CORBUF(I)="   "
61:       END FOR
62:       JCNT=1
63:       LOOP
64:       .  DO
65: 1     .  .  BUFFER IN(INFILE,FHDR,B,260,IFSTAT,IFLEN)
66:       .  .  CALL STATUS(INFILE)
67:       .  .  EXIT LOOP IF(IFSTAT.GT.3)
68:       .  UNTIL (IFLEN.EQ.256)
69:       .  CALL CNVNEG(FHDR,256)
70:       .  IRN=FHDR(1)
71:       .  ISN=FHDR(3)
72:       .  KTYP=FHDR(2)
73:       .  FOR J=1,7
74:       .  .  TRAY(J)=FHDR(J+27)
75:       .  END FOR
76:       .  CALL ITMCNT
77:       .  SBT=CSEC
78:       .  SD=FHDR(26)/1000.
79:       .  RD=FHDR(19)/1000.
80:       .  SIZE=(10.**(FHDR(36)/1000.))/1000.
```

```
 81:       .  RANGE=FLOAT(FHDR(11))*1000.+FLOAT(FHDR(12))+FLOAT(FHDR(13))/1000.
 82:       .    IF(RANGE.LT.0.0) CALL RANGER(ILAT,ILON,ELAT,ELON,RANGE)
 83:       .  A(1)=FLOAT(FHDR(16))/1000.
 84:       .  A(2)=FLOAT(FHDR(18))/1000.
 85:       .  A(3)=FLOAT(FHDR(22))/1000.
 86:       .  A(4)=FLOAT(FHDR(24))/1000.
 87:       .  ILAT =ISIGN(1,FHDR(15))*(ABS(FLOAT(FHDR(15)))+ABS(A(1)))
 88:       .  ILON =ISIGN(1,FHDR(17))*(ABS(FLOAT(FHDR(17)))+ABS(A(2)))
 89:       .  ELAT =ISIGN(1,FHDR(21))*(ABS(FLOAT(FHDR(21)))+ABS(A(3)))
 90:       .  ELON =ISIGN(1,FHDR(23))*(ABS(FLOAT(FHDR(23)))+ABS(A(4)))
 91:       .  SAMP=FHDR(38)
 92: C MAKE CORFILE
 93:       .  ENCODE(60,8080,CORBUF)IRN,ISN,TRAY(1),JULD,
 94:      +.  (TRAY(J),J=4,7)
 95: 8080  .  FORMAT(2I6,"01",I4,I3,3I2,I3,"01",28X)
 96:       .  ENCODE(60,8090,CORBUF(21))SD,SIZE,RD,RANGE
 97: 8090  .  FORMAT(10X,2F10.4,10X,2F10.4)
 98:       .  ENCODE(121,8100,CORBUF(74))ELAT,ELON,ILAT,ILON
 99: 8100  .  FORMAT(1X,4(F10.4,20X))
100:       .  BUFFER OUT(CORFIL,CORBUF,B,140,ICSTAT,ICLEN)
101:       .  CALL STATUS(CORFIL)
102:       .  IF (HDRSW)
103:       .  .  LIM=FHDR(39)
104:       .  .  FOR I=1,LIM
105:       .  .  .  LFN=60+I
106:       .  .  .  OPEN LFN
107:       .  .  .  CALL BUFOUT(LFN,MBUF,112,IEOF)
108:       .  .  .  CALL BUFOUT(LFN,MBUF,112,IEOF)
109:       .  .  .  CALL BUFOUT(LFN,MBUF,112,IEOF)
110:       .  .  .  ENDFILE LFN
111:       .  .  .  INEXT(I)=4
112:       .  .  END FOR
113:       .  .  IRNUM=0
114:       .  .  HDRSW=.FALSE.
115:       .  END IF
116:       .  FOR J=1,7
117:       .  .  TRAY(J)=FHDR(J+3)
118:       .  END FOR
119:       .  CALL ITMCNT
120:       .  DST=CSEC
121: C ENCODE INFO FROM ROSE FORMAT FILE HEADER INTO HIG DEMUX FILE HEADER
122:       .  IF(INUM.GE.200.AND.INUM.LE.299)
123:       .  .  SAMP=(SAMP/100)+1
124:       .  END IF
125:       .  ENCODE(51,8070,HDREC)INUM,ENUM,SBT,DST,SAMP
126: 8070  .  FORMAT(2I6,2X,2I16,I3,2X)
127:       .  BUFFER OUT(HDFIL,HDREC,B,17,MSTAT,MLEN)
128:       .  CALL STATUS(HDFIL)
129: C     NOW TO COPY EACH COMPONENT
130: C
131:       .  IRNUM=IRNUM+1
132:       .  FOR I=1,LIM
133:       .  .  IBASE=60+20*(I-1)
134:       .  .  IF(IBASE.GT.240)
135:       .  .  .  IBASE=240
136:       .  .  END IF
137:       .  .  WDSLFT=(FHDR(IBASE+11)-1)*4096+FHDR(IBASE+12)
138:       .  .  IF(FHDR(IBASE+11).EQ.0)
139:       .  .  .  WDSLFT=FHDR(71)*4096
140:       .  .  END IF
141:       .  .  MAXBUF=MAX0(MAXBUF,WDSLFT)
142:       .  .  LFN=60+I
143:       .  .  IRECNO(IRNUM,I)=INEXT(I)
144:       .  .  INWDS=0
145:       .  .  IPTR=1
146:       .  .  LOOP
147:       .  .  .  FOR J=1,4032
148:       .  .  .  .  IF (IPTR.GT.INWDS)
149:       .  .  .  .  .  INWDS=MIN0(4096,WDSLFT)
150:       .  .  .  .  .  EXIT FOR IF(INWDS.EQ.0)
151:       .  .  .  .  .  BUFFER IN(INFILE,IBUF,B,INWDS,ISTAT,ILEN)
152:       .  .  .  .  .  CALL STATUS(INFILE)
153:       .  .  .  .  .  CALL CNVNEG(IBUF,INWDS)
154:       .  .  .  .  .  WDSLFT=WDSLFT-INWDS
155:       .  .  .  .  .  IPTR=1
156:       .  .  .  .  END IF
157:       .  .  .  .  MBUF(J)=IBUF(IPTR)
158:       .  .  .  .  IPTR=IPTR+1
159:       .  .  .  END FOR
160:       .  .  .  J=J-1
```

```
161:          . . .  CALL BUFOUT(LFN,MBUF,J,IEOF)
162:          . . .  EXIT LOOP IF (INWDS.EQ.0)
163:          . .  END LOOP
164:          . .  ENDFILE LFN
165:          . .  CALL DSTAT(LFN,STATS,INEXT(I))
166:          .  END FOR
167:          END LOOP
168:          ENDFILE HDFIL
169:          IRNUM=0
170:          REWIND HDFIL
171:          LOOP
172:          .  BUFFER IN(HDFIL,HDREC,B,17,ISTAT,ILEN)
173:          .  CALL STATUS(HDFIL)
174:          .  EXIT LOOP IF(ISTAT.GE.3)
175:          .  IRNUM=IRNUM+1
176:          .  FOR I=1,LIM
177:          . .  LFN=60+I
178:          . .  HDREC(18)=IRECNO(IRNUM,I)
179:          . .  ENCODE(2,8110,HDREC(5))I
180: 8110     . .  FORMAT(I2)
181:          . .  CALL BUFOUT(LFN,HDREC,112,IEOF)
182:          .  END FOR
183:          END LOOP
184:          FOR I=1,112
185:          .  MBUF(I)=0
186:          END FOR
187:          FOR I=1,LIM
188:          .  LFN=60+I
189:          .  END FILE LFN
190:          .  CALL DSTAT(LFN,STATS,ILAST)
191:          .  CALL DPOS(LFN,2)
192:          .  MBUF(112)=INEXT(I)
193:          .  MBUF(111)=MAXBUF
194:          .  CALL BUFOUT(LFN,MBUF,112,IEOF)
195:          .  CALL DPOS(LFN,ILAST)
196:          .  CLOSE LFN
197:          END FOR
198:          STOP PAU
199:          END
200:            SUBROUTINE RANGER(RLAT,RLON,SLAT,SLON,RAN)
201:          DR=3.141592654/180.
202:          DY=RLAT-SLAT
203:          DX=(RLON-SLON)*COS((RLAT+SLAT)*DR/2.)
204:          RAN=1.852*SQRT(DX*DX+DY*DY)*60.
205:            RETURN
206:          END
```

```
 1:       SUBROUTINE ITMCNT,CNTITM,LJLCNT
 2:       INTEGER TRAY(7),YR,MO,DA,HR,MN,SC,MS
 3:       COMMON /ITCM/TRAY,CSEC,JULD
 4:       INTEGER*6 CSEC,TEMP
 5:       EQUIVALENCE (TRAY(1),YR),(TRAY(2),MO),(TRAY(3),DA),
 6:      +(TRAY(4),HR),(TRAY(5),MN),(TRAY(6),SC),(TRAY(7),MS)
 7: C
 8: C     TMCENT CONVERTS TIME TO CENTURY SEC AND MILLISECOND
 9: C     CENTTM CONVERTS CENTURY SEC TO REGULAR TIME.
10: C     THESE ROUTINES GOOD FOR ALL OF 1900'S AND UP
11: C
12:       JULD=0
13:       ENTRY LJLCNT
14:       IF (YR.GT.99)
15:       .  YR=YR-1900
16:       END IF
17:       IDAY=(1461*YR-1)/4
18:       MODYR=MOD(YR,4)
19:       IF (MODYR.EQ.0)
20:       .  MODYR=1
21:       ELSE
22:       .  MODYR=0
23:       END IF
24:       IF (JULD.EQ.0)
25:       .  JDAY=30.55*(MO+2)
26:       .  IF (MO.GT.2)
27:       .  .  JDAY=JDAY-93+MODYR
28:       .  ELSE
29:       .  .  JDAY=JDAY-91
30:       .  END IF
31:       .  JULD=JDAY+DA
32:       END IF
33:       IDAY=IDAY+JULD
34:       CSEC=IDAY*86400.0
35:       CSEC=CSEC+3600.0*HR+60.0*MN+SC*1.0
36:       TEMP=1000
37:       CSEC=CSEC*TEMP
38:       TEMP=MS
39:       CSEC=CSEC+TEMP
40:       YR=YR+1900
41:       RETURN
42: C
43: C     CODING FOR CENTTM
44: C
45:       ENTRY CNTITM
46:       TEMP=MOD2(CSEC,1000D)
47:       MS=TEMP
48:       TEMP=CSEC/1000D
49:       FSEC=TEMP
50:       IDAY=AMOD(FSEC,86400.0)
51:       HR=IDAY/3600
52:       IDAY=MOD(IDAY,3600)
53:       MN=IDAY/60
54:       SC=MOD(IDAY,60)
55:       IDAY=FSEC/86400.0
56:       YR=IDAY/365.25
57:       IDAY=IDAY-((1461*YR-1)/4)
58:       JULD=IDAY
59:       MODYR=MOD(YR,4)
60:       IF (MODYR.EQ.0)
61:       .  MODYR=1
62:       ELSE
63:       .  MODYR=0
64:       END IF
65:       IF (IDAY.GT.(59+MODYR))
66:       .  JDAY=IDAY+93-MODYR
67:       ELSE
68:       .  JDAY=IDAY+91
69:       END IF
70:       MO=(JDAY/30.55)
71:       DA=JDAY-IFIX(MO*30.55)
72:       MO=MO-2
73:       YR=YR+1900
74:       RETURN
75:       END
```

```
1:        SUBROUTINE CNVNEG(IBUF,LEN)
2:        INTEGER IBUF(4096)
3:        FOR I = 1,LEN
4:        IF (IBUF(I).GE.32768)
5:        IBUF(I) = IBUF(I) - 65536
6:        ENDIF
7:        ENDFOR
8:        RETURN
9:        END
```

```
1:        SUBROUTINE BUFOUT(LFN,OUT,M,K,N), BUFIN(LFN,OUT,M,K,N)
2: C SHORTENS CODING FOR BUFFER IN/OUT WORK
3:        DIMENSION OUT(1)
4:        BUFFER OUT(LFN,OUT,B,M,K,N)
5: 1      CALL STATUS(LFN)
6:        GOTO (1,2,2) ,K
7: 2      RETURN
8:        ENTRY BUFIN
9:        BUFFER IN(LFN,OUT,B,M,K,N)
10: 3     CALL STATUS(LFN)
11:       GOTO (3,4,4) ,K
12:       RETURN
13:       END
```

::
::

Program notes:

Harris structured FORTRAN 66 and 77 are the versions of FORTRAN language used throughout these programs. The user callable library subroutines DATE, TIME, BTIME and ETIME are described in the FORTRAN Reference Manual 0861004-003 sections 9.3.3, 9.3.7 and 9.3.30. Statements BUFFER IN and BUFFER OUT with the associated library subroutine STATUS are used throughout to permit records of arbitrary length and format to be read and written asynchronously. Detailed descriptions can be found in sections 6.13 and 9.3.27 of the manual.