MANAGEMENT OF A
SIGNAL MEASUREMENT DATA BASE (SMDB)

TECHNICAL REPORT NO. 2

PREPARED BY
DONALD L. DIETZ AND JEFFREY S. SHAUB

PREPARED FOR
AIR FORCE TECHNICAL APPLICATIONS CENTER
ALEXANDRIA, VIRGINIA 22314

ENSCO, INC.

SEISMIC APPLIED RESEARCH DIVISION
813 NORTH ROYAL STREET
ALEXANDRIA, VIRGINIA 22314

411669
This report describes a suite of programs to be used for management of a Signal Measurement Data Base (SMDB). The programs permit a user to write on disk, from tape, files of seismic signal measurements which have been used, for example, in VELA-sponsored identification experiments. The program suite consists of software to establish the SMDB and to enable user interface with the data base. Moreover, the structure of the
20. (continued)

SMDB is also presented here to illustrate its expandable and maintainable format.
This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by AFTAC/VSC, Patrick Air Force Base, FL 32925, under Contract Number F08606-79-C-0014.

AFTAC Project Number: VELA T/9705/B/PMP

Project Title: VELA Network and Automatic Processing Research

ARPA Order Number: 2551

Name of Contractor: ENSCO, Incorporated

Contract Number: F08606-79-C-0014

Effective Date of Contract: 11 December 1978

Contract Expiration Date: 31 March 1980

Project Manager: Theodore J. Cohen (703) 548-8666
SUMMARY

This report describes a suite of programs to be used for management of a Signal Measurement Data Base (SMDB). The programs permit a user to write on disk, from tape, files of seismic signal measurements which have been used, for example, in VELA-sponsored identification experiments. The program suite consists of software to establish the SMDB and to enable user interface with the data base. Moreover, the structure of the SMDB is also presented here to illustrate its expandable and maintainable format.

Neither the Advanced Research Projects Agency nor the Air Force Technical Applications Center will be responsible for information contained herein which has been supplied by other organizations or contractors, and this document is subject to later revision as may be necessary. The views and conclusions presented are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Projects Agency, the Air Force Technical Applications Center, or the US Government.

ENS CO, INC.
<table>
<thead>
<tr>
<th>SECTION</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SUMMARY</td>
<td>ii</td>
</tr>
<tr>
<td>I.</td>
<td>INTRODUCTION</td>
<td>I-1</td>
</tr>
<tr>
<td>II.</td>
<td>STRUCTURE OF THE SIGNAL MEASUREMENT DATA BASE (SMDB)</td>
<td>II-1</td>
</tr>
<tr>
<td>III.</td>
<td>ESTABLISHMENT OF THE SMDB</td>
<td>III-1</td>
</tr>
<tr>
<td>IV.</td>
<td>CONCLUSIONS AND RECOMMENDATIONS</td>
<td>IV-1</td>
</tr>
<tr>
<td>V.</td>
<td>REFERENCES</td>
<td>V-1</td>
</tr>
<tr>
<td>Appendix A</td>
<td>DOCUMENTATION OF SIGNAL MEASUREMENT DATA BASE (SMDB) ROUTINES</td>
<td>A-1</td>
</tr>
<tr>
<td>Appendix B</td>
<td>LISTINGS OF SIGNAL MEASUREMENT DATA BASE (SMDB) PROGRAMS</td>
<td>B-1</td>
</tr>
<tr>
<td>Appendix C</td>
<td>FORMAT DESCRIPTION OF ENSCO'S RAW SIGNAL MEASUREMENT TAPE</td>
<td>C-1</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>II-1</td>
<td>ORGANIZATION OF THE SIGNAL MEASUREMENT DATA BASE (SMDB)</td>
<td>II-2</td>
</tr>
<tr>
<td>II-2</td>
<td>DIRECTORY ENTRIES</td>
<td>II-4</td>
</tr>
<tr>
<td>II-3</td>
<td>STRUCTURE OF DATA RECORD IN SIGNAL MEASUREMENT DATA FILE</td>
<td>II-7</td>
</tr>
<tr>
<td>C-1</td>
<td>ORGANIZATION OF RAW SIGNAL MEASUREMENT TAPE</td>
<td>C-3</td>
</tr>
</tbody>
</table>
### LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>III-1</td>
<td>SIGNAL MEASUREMENT DATA HEADER</td>
<td>III-2</td>
</tr>
<tr>
<td>C-1</td>
<td>SITE NUMBER ASSIGNMENTS</td>
<td>C-4</td>
</tr>
<tr>
<td>C-2</td>
<td>STANDARD EVENT HEADER DESCRIPTION</td>
<td>C-6</td>
</tr>
<tr>
<td>C-3</td>
<td>RAW SIGNAL MEASUREMENTS</td>
<td>C-13</td>
</tr>
</tbody>
</table>
SECTION I
INTRODUCTION

The purpose of the Data Base Transfer (Task 4.3.1 under Contract Number F08606-79-C-0014) is to transfer the Event Discriminant Data Base from the PDP-15/50 to the PDP-11/70, and to do so under the following constraints:

- Preserve the signal measurements for all event-stations processed under Contract Number F08606-79-C-0014.

- Establish an expandable and maintainable Signal Measurement Data Base (SMDB) on the PDP-11/70 that:
  - satisfies the first constraint above
  - may be updated with additional signal measurements for event-station data processed by other contractors.

- Provide FORTRAN-compatible software utilities to:
  - initialize the SMDB Directory and Free-Block File
  - update the SMDB by event, station, measurement, or contractor
  - access the SMDB for information by event, station, measurement, or contractor.

- Write selected driving programs to demonstrate an:
  - SMDB update with signal measurements discussed in the first criterion
  - SMDB access to list Directory information and signal measurements
- SMDB access to compute unbiased network averages of signal measurements for a specified event.

- Demonstrate this software contingent on the availability of UNIX operating system utilities provided by the Government.

Section II of the report describes the detailed structure of the SMDB. Section III discusses the establishment of the SMDB, utilizing signal measurements for all event-station data processed under Contract Number F08606-79-C-0014. Section IV presents conclusions and recommendations for future work. Finally, documentation of the software utilities developed during the task is provided in Appendix A of this report, while listings of the software may be found in Appendix B. Appendix C contains a description of ENSCO's raw signal measurement tape format.
SECTION II
STRUCTURE OF THE SIGNAL MEASUREMENT DATA BASE (SMDB)

The Signal Measurement Data Base (SMDB) uses a multi-file structure which is designed and implemented in FORTRAN IV for the PDP-11/70. The principal items comprising the SMDB are vectors of signal measurements obtained by preprocessing waveform data from various recording stations. The design of the data base was predicated on the following criteria:

- Expandability with respect to events, stations, and measurements.
- Allocations for data from up to four contractors.

To this end, a linklist structure was employed in organizing the data.

The SMDB consists of four types of files:

- A Directory File (SIGMS.DIR).
- A Directory Freeblock File (SIGMSDIR.FRE).
- Signal Measurement Data Files (one file for each event).
- Freeblock File for Data Files (one file for each event).

The general organization of the SMDB is shown in Figure II-1.

The directory file, SIGMS.DIR, consists of a header record followed by Event Base Vectors and Station Entry Vectors. These vectors are for sorting and retrieving event/
FIGURE 11-1

ORGANIZATION OF THE SIGNAL MEASUREMENT DATA BASE (SMDB)

Note: A signal measurement data file and its associated free-block file exist for each event. File names are as shown with the event designation number substituted for the characters XXXX.

Directory File 'SIGNS.DIR'

Freeblock File For Data File 'SIGMXXX.FRE'

Signal Measurement Data File 'SIGMXXX.DAT'

Freeblock File For Data File 'SIGMXXX.FRE'

Directory File 'SIGNS.DIR'

Signal Measurement Data File 'SIGMXXX.DAT'

Freeblock File For Data File 'SIGMXXX.FRE'

Directory File 'SIGNS.DIR'

Signal Measurement Data File 'SIGMXXX.DAT'

Freeblock File For Data File 'SIGMXXX.FRE'

Directory File 'SIGNS.DIR'

Signal Measurement Data File 'SIGMXXX.DAT'

Freeblock File For Data File 'SIGMXXX.FRE'

Directory File 'SIGNS.DIR'

Signal Measurement Data File 'SIGMXXX.DAT'

Freeblock File For Data File 'SIGMXXX.FRE'
signal information. The header record contains the following information:

- The current number of events in the system.
- The record number of the first event entry.
- The current number of directory records that are in use.
- The maximum number of directory records allowed.

All entries are allocated four bytes of storage.

The structure of the Event Base Vector and Station Entry Vector are shown in Figure II-2. Each entry record is 128 bytes long and consists of thirty-two four-byte words. The attributes of the Event Base Vector are defined as follows:

- Event Sequence Number - A virtual integer J, where 1≤J≤# of events, which indicates the order an event appears in the directory. For example, the fifth event listed in the directory has an event sequence number of 5. The sequence number for an event changes when the SMDB is updated with new events.
- Event Designation Number - A unique four-digit number appearing as the last four characters of the ENSCO-alphanumeric event designation.
- Event Origin Time - Source time in YY DDD HH MM SS (5 word integer format).
- Event Latitude - Source latitude in degrees, N(+), S(-).
- Event Longitude - Source longitude in degrees, E(+), W(-).
- Event mb - Event bodywave magnitude.

ENSCO, INC. II-3
### EVENT BASE VECTOR

<table>
<thead>
<tr>
<th>Event Sequence Number (Virtual)</th>
<th>Event Designation Number</th>
<th>Event Origin Time</th>
<th>Event Latitude +N°</th>
<th>Event Longitude +E°</th>
<th>Event mn</th>
<th>Event Depth</th>
<th>Unused (22*4 words)</th>
<th>Backward Pointer To Last Event</th>
<th>Forward Pointer To Next Event</th>
<th>Total Number Of Stations</th>
<th>Pointer To First Station</th>
</tr>
</thead>
</table>

### STATION ENTRY VECTOR

<table>
<thead>
<tr>
<th>Station Name</th>
<th>Station Number</th>
<th>Unused (15*4 words)</th>
<th>Backward Pointer To Last Station</th>
<th>Forward Pointer To Next Station</th>
<th>Total Number Of Contractors</th>
<th>Contractor Name</th>
<th>First Measurement Record</th>
<th>Number Of Measurements</th>
</tr>
</thead>
</table>

### FIGURE II-2

DIRECTORY ENTRIES
• Event Depth - Source depth in km.

• Backward Pointer to Last Event - The record number corresponding to the previous event base vector (zero if first event).

• Forward Pointer to Next Event - The record number corresponding to the next event base vector (zero if last event).

• Total Number of Stations - The total number of station entry vectors for the event. This corresponds to the total number of unique stations for which measurements are available for a given event.

• Point to First Station - The record number corresponding to the first station entry vector for the event.

The attributes of the Station Entry Vector are defined as follows:

• Station Name - The four-character ENSCO-alphanumeric station designation.

• Station Number - The integer number $K$, $1 \leq K \leq 50$, assigned to a given station.

• Backward Pointer to Last Station - The record number corresponding to the last station entry vector (zero if first station).

• Forward Pointer to Next Station - The record number corresponding to the next station entry vector (zero if last station).

• Total Number of Contractors - The total number of contractors contributing signal measurements for an event-station (maximum of four).
Contractor Name - The four-character alphanumeric designation for a contributor to the database.

First Measurement Record - The record number in the signal measurement data file corresponding to the first record of signal measurements.

Number of Measurements - The total number of measurement values contributed by the associated contractor.

Both the event base vectors and the station entry vectors have unused locations where, if desired, additional information can be stored.

The directory freeblock file, SIGMSDIR.FRE, consists of a header record containing the current number of non-sequential free records in the directory file. This header record is followed by records consisting of 256 four-byte words. These records each contain up to 256 record numbers. The function of this freeblock is to allow the reuse of records freed by the deletion of entries in the directory file. This practice minimizes the size of the directory file.

Each event has one signal measurement file with a name of the form SIGMXXXX.DAT, where XXXX is the event designation number. The records in these files consist of 64 four-byte words. Two types of records make up the data file: a header record and data records (multiple). The header record contains the current number of data records in use and the maximum number of data records allowed in the first two positions. The structure of the data records is shown in Figure II-3. The major attributes are defined as follows:
| 31 Measurement Values (4 * byte word) | 31 Measurement Labels (4 * byte word) | Last Measurement Record | Next Measurement Record |

(file name of form 'SIMGXXXX.DAT' where XXXX=Event Designation Number)

**FIGURE II-3**

STRUCTURE OF DATA RECORD IN SIGNAL MEASUREMENT DATA FILE
• Measurement Value - A signal parameter obtained by preprocessing waveform data from various recording stations. In the present case, the measurements are ordered into four categories depending on the type of data from which the discriminants are derived (namely, long-period signals/noise, short-period regional signals/noise, short-period teleseismic signals/noise, and short-period signals/noise).

• Measurement Labels - A four-character alphanumeric designation associated with each measurement value. This label is used as a keyword in locating a given signal parameter.

• Last Measurement Record - The record number corresponding to the last data record associated with a given contractor (zero if first record).

• Next Measurement Record - The record number corresponding to the next data record associated with a given contractor (zero if last record).

Each event also has a freeblock file associated with its signal measurement file. The files are named analogously to the data file. The file name is of the form SIGMXXXX.FRE, where XXXX is the event designation number. The structure and function of this file are the same as those of the directory freeblock file.
SECTION III
ESTABLISHMENT OF THE SMDB

The SMDB is established by executing DDBASE. This routine takes a labeled 1600 bpi 9-track signal measurement tape as input. The data headers for each event-station file should follow the description in Table III-1*. Before DDBASE is initially run, the program INITIAL must be executed. This program initializes the directory files of the data base and deletes any data files that may be present. In general, the UNIX command SMT must also be performed with the appropriate options before any execution of DDBASE. These routines along with other support routines and subroutines are documented in Appendix A. The source file names and the names of the associated source files and libraries (if any) are also provided. The algorithms can be executed by following the standard UNIX FORTRAN guidelines. Appendix B contains compiled listings of the data base management and demonstration programs.

* The conversion of the data header from the standard ENSCO format (see Appendix C) is performed by the IBM 360/44 program COPY99.
TABLE III-1
SIGNAL MEASUREMENT DATA HEADER
(PAGE 1 OF 3)

<table>
<thead>
<tr>
<th>Position</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I</td>
<td>Seismogram number (not used).</td>
</tr>
<tr>
<td>2</td>
<td>I</td>
<td>Number of components (always equal to one).</td>
</tr>
<tr>
<td>3</td>
<td>I</td>
<td>Edit length (always equal to 100).</td>
</tr>
<tr>
<td>4</td>
<td>F</td>
<td>Sample rate (not used).</td>
</tr>
<tr>
<td>5</td>
<td>F</td>
<td>Edit start time: hours</td>
</tr>
<tr>
<td>6</td>
<td>F</td>
<td>Edit start time: minutes</td>
</tr>
<tr>
<td>7</td>
<td>F</td>
<td>Edit start time: seconds</td>
</tr>
<tr>
<td>8-9</td>
<td>A</td>
<td>'TIHEADER'.</td>
</tr>
<tr>
<td>10-12</td>
<td>A</td>
<td>Event designation (word 12 is the event designation number).</td>
</tr>
<tr>
<td>13-14</td>
<td>A</td>
<td>Data type ('DISCR').</td>
</tr>
<tr>
<td>15</td>
<td>A</td>
<td>Data orientation code ('RAW').</td>
</tr>
<tr>
<td>16-47</td>
<td>A</td>
<td>Site status table (not used).</td>
</tr>
<tr>
<td>48</td>
<td>A</td>
<td>'bbYY'</td>
</tr>
<tr>
<td>49</td>
<td>A</td>
<td>'bDDD'</td>
</tr>
<tr>
<td>50</td>
<td>A</td>
<td>'bbHH' event origin time.</td>
</tr>
<tr>
<td>51</td>
<td>A</td>
<td>'bbMM'</td>
</tr>
<tr>
<td>52</td>
<td>A</td>
<td>'bbSS'</td>
</tr>
<tr>
<td>53</td>
<td>A</td>
<td>Confidence of source time (PDE code) (not used).</td>
</tr>
<tr>
<td>54-55</td>
<td>A</td>
<td>Event latitude (±90°N) (represents F8.3).</td>
</tr>
<tr>
<td>56-57</td>
<td>A</td>
<td>Event longitude (±180°E) (represents F8.3).</td>
</tr>
<tr>
<td>58</td>
<td>A</td>
<td>Event depth (represent integer).</td>
</tr>
<tr>
<td>59</td>
<td>A</td>
<td>Event mb (represents F4.2).</td>
</tr>
</tbody>
</table>
### TABLE III-1
SIGNAL MEASUREMENT DATA HEADER
(PAGE 2 OF 3)

<table>
<thead>
<tr>
<th>Position</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>A</td>
<td>Event $M_s$ (represents F4.2).</td>
</tr>
<tr>
<td>61</td>
<td>A</td>
<td>Event $m_b$ (represents F4.2).</td>
</tr>
<tr>
<td>62</td>
<td>A</td>
<td>ENSCO estimated $M_s$ (represents F4.2).</td>
</tr>
<tr>
<td>63-64</td>
<td>A</td>
<td>Information source (e.g., 'NEIL').</td>
</tr>
<tr>
<td>65</td>
<td>A</td>
<td>Event-station tectonic class code.</td>
</tr>
<tr>
<td>66</td>
<td>A</td>
<td>Not used.</td>
</tr>
<tr>
<td>67</td>
<td>A</td>
<td>Site number (represents integer).</td>
</tr>
<tr>
<td>68-69</td>
<td>A</td>
<td>Signal measurement fields for specific phase (represents F8.3) (meaningless in context of signal measurement tape).</td>
</tr>
<tr>
<td>96-97</td>
<td>A</td>
<td>Signal measurement fields for specific phase (represents F8.3) (meaningless in context of signal measurement tape).</td>
</tr>
<tr>
<td>98</td>
<td>A</td>
<td>Array name ('SROb').</td>
</tr>
<tr>
<td>99-157</td>
<td>A</td>
<td>Not used.</td>
</tr>
<tr>
<td>158</td>
<td>A</td>
<td>Array name.</td>
</tr>
<tr>
<td>159-160</td>
<td>A</td>
<td>Site latitude (+90°N) (represents F8.3).</td>
</tr>
<tr>
<td>161-162</td>
<td>A</td>
<td>Site longitude (+180°E) (represents F8.3).</td>
</tr>
<tr>
<td>163</td>
<td>A</td>
<td>P-wave arrival seconds into edit (represents integer).</td>
</tr>
<tr>
<td>164</td>
<td>A</td>
<td>S-wave arrival seconds into edit (represents integer).</td>
</tr>
<tr>
<td>165</td>
<td>A</td>
<td>LQ-wave arrival seconds into edit (represents integer).</td>
</tr>
<tr>
<td>166</td>
<td>A</td>
<td>LR-wave arrival seconds into edit (represents integer).</td>
</tr>
<tr>
<td>Position</td>
<td>Data Type</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>167</td>
<td>A</td>
<td>Estimated LR length (seconds) (represents integer).</td>
</tr>
<tr>
<td>168-169</td>
<td>A</td>
<td>Station azimuth (degrees) (great circle beam direction) (represents F8.3).</td>
</tr>
<tr>
<td>170</td>
<td>A</td>
<td>Station elevation (meters) (great circle beam direction) (represents integer).</td>
</tr>
<tr>
<td>171-172</td>
<td>A</td>
<td>Source-to-station epicentral distance great circle distance (degrees) (represents F8.3).</td>
</tr>
<tr>
<td>173-322</td>
<td>A</td>
<td>Not used.</td>
</tr>
<tr>
<td>323-324</td>
<td>A</td>
<td>Signal measurement fields for specific phases (represents F8.3) (meaningless in context of signal measurement tape).</td>
</tr>
<tr>
<td>373-374</td>
<td>A</td>
<td>Signal measurement fields for specific phases (represents F8.3) (meaningless in context of signal measurement tape).</td>
</tr>
<tr>
<td>375</td>
<td>A</td>
<td>Not used.</td>
</tr>
</tbody>
</table>
SECTION IV
CONCLUSIONS AND RECOMMENDATIONS

An expandable and maintainable Signal Measurement Data Base (SMDB) program suite has been designed and compiled on the PDP-11/70 under the UNIX operating system. Tapes (from the SDAC) which were used in the VELA-sponsored identification experiment, and which were used in the development and test of the SMDB program suite, are:

1. L22882 (IBM 360/44)
2. L13033 (DEC PDP-15/50).

These tapes can be used in future efforts to refine the SMDB program suite.

It should be noted that several problems were encountered with the UNIX FORTRAN compilers during attempts to implement the SMDB routines. Specifically, the routines were initially coded in a modified version of DEC RT-11 FORTRAN IV (Rottman, 1975). (The UNIX command to request this compiler is FORTRAN.) Difficulties encountered with this compiler included problems which are apparently related to array size and to the argument lists. That is, a routine could be made to generate routine errors, or not to do so, by increasing or decreasing the size of its arrays, respectively. Associated error messages, in general, indicated a bus error or a segmentation violation. Regardless, a solution to this problem was not readily apparent. Further, the number of
arguments in a subroutine argument list appeared to be limited to eight. Any additional arguments were not passed. It was possible to circumvent this problem by passing arguments in a named common block.

Toward the end of this task, a new FORTRAN compiler was provided (Anon., 1978). This compiler is based on American National Standard (ANSI) FORTRAN 77. (The UNIX command to request this compiler is F77.) An investigation was made into the feasibility of converting the SMDB routines to the FORTRAN 77 standard. Necessary changes would include substitution of the ENCODE/DECODE statements with equivalent statements, modification of the file I/O, modification of character variables, and modification of the code to allow for irregularities in the new compiler. ENCODE/DECODE statements can be replaced with 'reads' and 'writes' to internal files. This replacement is not possible on a one-to-one basis, and some additional program structuring will be necessary to provide for parallel operations. The file I/O initiated using the routines SETFIL and DFILE does not appear to be equivalent to using SETFIL and DEFINE FILE in the original compiler. In particular, it seems that the file to be accessed must already exist in order to be accessed (i.e., the system does not create a new file). This was not the case with the compiler called using FORTRAN. Another difference in file I/O requirements was that the last record written in a file could not be read subsequent to the write to the file. This difficulty may indicate that it is necessary to close and open the file between write and read operations. An example of one irregularity encountered during the investigation of the F77 compiler is the failure to
compile a logical expression with a unary minus (e.g., if (IFLG.EQ.-1) go to 100). The same logical expression without the minus sign, however, does compile.

The results of the investigation made indicate that the F77 compiler is sufficiently different from the FORTRAN compiler as to require a significant effort to convert the SMDB from the FORTRAN to the F77 compiler. It will first be necessary to establish the operating characteristics of the new compiler (i.e., the logic sequences necessary to accomplish a given operation). After the methods for performing the desired operations have been established (usually done by the trial-and-error method), the SMDB routines can be restructured (i.e., converted).

At this time, a competent FORTRAN language compiler is not available on the SDAC PDP-11/70. Note, however, that the use of the FORTRAN language is not recommended by the designers of the UNIX operating system for major applications using UNIX.

As a final comment, several additional routines could be added to make the SMDB more versatile. These include a function to delete event and/or station entries and a function to modify individual measurements resident in the database. The design of the SMDB included consideration of these avenues for development. Several of the routines documented in Appendix A could be used to implement these options.

Kerningham, B., undated; UNIX For Beginners, Bell Laboratories, Murray Hill, NJ.

Ringdal, F., 1975; Maximum Likelihood Estimation of Seismic Event Magnitudes from Network Data, Technical Report No. 1, Texas Instruments Report No. ALEX(01)-TR-75-01, AFTAC Contract Number F08606-75-C-0029, Texas Instruments Incorporated, Dallas, TX.

Rottman, J. N., 1975 (revised); A Guide to the UNIX FORTRAN System, Princeton University, Princeton, NJ.

APPENDIX A
DOCUMENTATION OF SIGNAL MEASUREMENT DATA BASE (SMDB) ROUTINES

This appendix contains descriptions of the main routines and associated subroutines used to manage and demonstrate the SMDB. These descriptions include definitions of arguments, commons, and data types. Comments on the use of the routines are also provided. Applicable source files resident on the PDP-11/70 system are listed.

These routines can be incorporated into additional algorithms to manipulate the SMDB. Several examples of this type of application are generation of files of unbiased network magnitude measurements, computation of discriminants, and classification of events.
Subroutine:

ACCESS

Arguments:

(NDSG, STAT, CONTR, MSLAB, NRET, RMEAS, RLAB, IEVR).

NDSG - integer * 4
STAT - real * 4
CONTR - real * 4
MSLAB - real * 4
NRET - integer * 4
RMEAS - real * 4 array of length > NRET
RLAB - real * 4 array of length > NRET
IEVR - integer * 4

Commons:

COMMON/FLAGS/IFLGM, IFLGEC
IFLGM - integer * 2
IFLGEN - integer * 2
COMMON/ASVAR/NXREC1, NXREC2, NXREC3, NXREC4

Description:

ACCESS accesses the signal measurement data base and re-
turns the measurements.

NDSG is an event designation number.
STAT is a four-character station name (if it is ****, the
given measurement is returned for all stations re-
porting the event).
CONTR is a four-character contractor name.
MSLAB is a four-character measurement label.
NRET is the number of measurements returned.
RMEAS contains the measurements to be returned.

ENSICO, INC. A-2
RLAB contains the measurement labels when IFLGMS=0. IEVR is the event record number in the directory file SIGMS.DIR.

\[
\text{IFLGM} = \begin{cases} 
0, & \text{returns all measurements for the given station (in RMEAS) with the associated labels (in RLAB) — WARNING: it should be noted that there is no guarantee that all of the measurements will be in order (or exist) across stations.} \\
1, & \text{return only indicated measurement for the given station.} 
\end{cases}
\]

\[
\text{IFLGEV} = \begin{cases} 
0, & \text{search for event} \\
1, & \text{use event record number in IEVR.} 
\end{cases}
\]

NXREC1, NXREC2, NXREC3, NXREC4 are the associated variables for the maximum of four data base files open at one time.

Restrictions:

1) The event data file is opened on logical unit number 2.

2) It is assumed that the directory file SIGMS.DIR is already open on logical unit number 1.

Source File:

access.for

Associated Source File:

search.for

ENSCO, INC. A-3
Subroutine:

CHKFRE

Arguments:

(LUNIT, IFLAG, NFB, IFREC).
LUNIT - integer * 2
IFLAG - integer * 2
NFB - integer * 4
IFREC - integer * 4

Commons:

COMMON/ASVAR/NXREC1, NXREC2, NXREC3, NXREC4
all variables integer * 2

Description:

CHKFRE checks the indicated freeblock file and either returns a free record number or extends the freeblock file. The file is modified accordingly and the total number of freeblocks before the call is returned.

LUNIT is the logical unit number associated with the freeblock file (in practice LUNIT=3 is the freeblock file for the directory and LUNIT=4 is the freeblock file for the event data file.

0, returns free record number in IFREC
IFLAG = 1, freeblock extended with record number in IFREC.

NFB is the number of freeblocks prior to a call.
IFREC is the free record number.
NXREC1, NXREC2, NXREC3, NXREC4 are the associated variables for all possible open data base files.
Restrictions:

1) The appropriate freeblock files must be open on logical unit LUNIT.

Source File:

chkfre.for
Main Routine:
COPY99

Input:
Signal measurement tape in standard ENSCO format.

Description:
COPY99 is an IBM 360/44 program used to convert the data headers of the signal measurement tape to the format expected by the PDP-11/70 program DDBASE and output the results to another tape. The output tape is input to the signal measurement data base management routines. Prior to execution of this program, the character string 'NSKIP' should be changed to the number of data files to skip on the input tape before starting the conversion and copy. The input tape is accessed on unit 8 and the output tape is accessed on unit 9.
Main Routine:

DDBASE

Input:

9-track signal measurement tape with appropriate EBCDIC data headers on logical unit 0.

Description:

DDBASE accesses the information on a labeled 1600 bpi 9-track signal measurement tape and performs the necessary conversions and formatting before providing the information to subroutine UPDATE. This routine includes a look-up table for station names based on the station number, a look-up table for measurement labels, EBCDIC to ASCII conversion, and IBM 360/44 to PDP-11/70 floating-point conversion.

Restrictions:

1) The 9-track input tape must have the correct EBCDIC header.
2) All information for an event on a tape must be consecutive.
3) UNIX command smt must be performed before execution of DDBASE with the appropriate options.
4) The algorithm currently requires the tape to be mounted on unit 0 (allowable units are 0-7).

Source File:

ddbase.for
Associated Source Files:
  update.for
  search.for
  chkfre.for

Program Source File:
  ddbasep.for

Associated libraries (in addition to normal FORTRAN library):
  /lib/f4u
Main Routine:

EVLIST

Input: (from terminal)

Logic flag (IFLG) (requested inputs 0 or 1) range of event sequence numbers (virtual).

Description:

EVLIST displays the event-station information stored in the data base in the form requested by the input parameters.

If a zero input for the logic flag the event information and the stations reporting are displayed.

If a one is input the individual station measurements for contractor ENSC are included.

Source File:

evlist.for

Associated Source Files:

search.for
access.for

Program Source File:

evlistp.for

ENSCO, INC. A-9
Main Routine:

INITIAL

Description:

INITIAL initializes the signal measurement data base. This is accomplished by deleting all existing event data files (if any) and by initializing the header records of the directory file (SIGMS.DIR) and its freeblock file (SIGMSDIR.FRE).

Program Source File:

initial.for
Subroutine:

MBIAS

Arguments:

(RMX, NSTA, NDISC, DISCR).
RMX - real * 4 of length > NSTA * NDISC.
NSTA - scalar (integer * 2).
NDISC - scalar (integer * 2).
DISCR - real * 4 array of length > NDISC.

Description:

MBIAS performs Ringdal's maximum likelihood estimation of event magnitude (Ringdal, 1975) using signal and noise measurements.

RMX is a matrix whose elements contain signal or noise measurements, or a flag indicating that the element is to be ignored.

\[ \text{signal measurement} + 1000 \]

\[ \text{noise measurement} - 1000 \]

\[ 0 \text{ if it is to be ignored.} \]

NSTA is the number of stations.

NDISC is the number of measurements per station.

DISCR contains the unbiased magnitude estimates.

Restrictions:

1) RMS contains the measurements for station 1 followed by the measurements for station 2, etc. until the measurements for station NSTA have been stored.

2) NSTA must be ≤ 50.

Source File:

mbiascat.for

ENSCO, INC. A-11
Main Routine:

NAVE and (NAVE2)

Input: (from terminal)

Event designation number.

Description:

NAVE and NAVE2 are drivers to demonstrate the subroutine MBIAS. The unbiased network averages are computed for several measurements for the designated event.

Source File:

nave.for and nave2.for

Associated Source Files:

mbiascat.for
search.for
access.for

Program Source File:

navep.for
Main Routine:
  RETURN

Input: (from terminal)
  Event designation number
  Station name
  Measurement label.

Description:
  RETURN is a debugging routine to test subroutines ACCESS and SEARCH. Various measurements are returned depending on the input parameters and the setting of flags hardwired into the program.

Source File:
  return.for

Associated Source Files:
  search.for
  access.for

Program Source File:
  returnp.for
Subroutine:
SEARCH

Arguments:

(IFLAG, KEY, EVREC, STREC, ICON, MSREC, MSIND, ISRCH)

IFLAG - integer * 2
KEY - four word real * 4 array
EVREC - integer * 4
STREC - integer * 4
ICON - integer * 4
MSREC - integer * 4
MSIND - integer * 2
ISRCH - integer * 2

Commons:
COMMON/ASVAR/NXREC2, NXREC2, NXREC3, NXREC4
all variables INTEGER * 2.

Description:

SEARCH searches the signal measurement data base for a match to a specific keyword and returns the appropriate record number and index (if applicable).

\[
\begin{align*}
\text{IFLAG} & = 0, \text{ full search for a given measurement (needs all keywords)} \\
& = 1, \text{ search for event designation number} \\
& = 2, \text{ search for station name (needs evrec)} \\
& = 3, \text{ search for contractor name (needs strec)} \\
& = 4, \text{ search for measurement (needs icon) (returns record number in 'sigmxxxx.det' file and the index into the record).}
\end{align*}
\]

ENSCO, INC. A-14
KEY is the four-word array containing the keywords

KEY(1) = event designation number (should be converted to floating point prior to call)
KEY(2) = four-character station name
KEY(3) = four-character contractor name
KEY(4) = four-character measurement label.

EVREC is the event record number in the directory file, SIGMS.DIR.

STREC is the station record number in the directory file, SIGMS.DIR.

ICON is the index into the station record of the contractor block - possible values are 21, 24, 27, and 30.

MSREC is the measurement record number in the event data file SIGMXXX.DAT.

MSIND is the index into the measurement record of the measurement value

ISRCH is an error flag

\[ ISRCH = \begin{cases} 0, & \text{successful search} \\ 1, & \text{unsuccessful search} \end{cases} \]

NXREC1, NXREC2, NXREC3, NXREC4 are the associated variables of all possible open data base files.

Restrictions:

1) If EVREC or STREC are equal to zero when IFLAG=2 or 3, respectively, the program is stopped. This also occurs if the number of stations or measurements is zero.

2) The content of KEY(1) must be in floating point representation.

3) The directory file SIGMS.DIR must be open on logical unit number 1 and the event data file must be open on logical unit number 2.
Source File:
search.for
Subroutine:
  UPDATE

Arguments:
  (IERR)
  IERR - integer * 2

Commons:
  COMMON/ARG/ESINFO
  ESINFO - 141-word real * 4 array
  common/ASVAR/NXREC1, NXREC2, NXREC3, NXREC4
  all variables integer * 2

Description:
  UPDATE establishes the signal measurement data base
given the pertinent event-station information and flags con-
tained in array ESINFO.
  IERR is an error flag.
    IERR = 0, no errors
    IERR = 1, error in establishing data base.
  ESINFO is the information packet passed to this data
  base routine.
  ESINFO(1) = event designation number (integer * 4).
  ESINFO(2) = bbYY
  ESINFO(3) = bDDD
  ESINFO(4) = bbHH \{ event origin time (integer * 4).
  ESINFO(5) = bbMM
  ESINFO(6) = bbSS
  ESINFO(7) = event latitude ($\pm$90°N)(represented as F8.3
               in this data base).
ESINFO(8) = event longitude (+180°E) (represented as F8.3 in this data base).
ESINFO(9) = event magnitude (m_b) (represented as F4.2 in this data base).
ESINFO(10) = depth of event (integer * 4).
ESINFO(11) = event flag IFLGEV (integer * 4).
ESINFO(12) = four-character station name.
ESINFO(13) = station number (integer * 4).
ESINFO(14) = station flag IFLGST (integer * 4).
ESINFO(15) = four-character contractor name.
ESINFO(16+(K-1)*2) = four-character measurement label K=1,62.
ESINFO(17+(K-1)*2) = associated measurement value K=1.62.
ESINFO(140) = event record number in the directory file SIGMS.DIR (integer * 4).
ESINFO(141) = station record number in the directory file SIGMS.DIR (integer * 4).
IFLGEV (ESINFO(11)) is a flag that controls the logic flow of this routine in regard to events.
   1, search for event
   0, add new event
   -1, update previous event (i.e., use event record number supplied in ESINFO(140)).
IFLGST (ESINFO(14)) is a flag that controls the logic flow of this routine in regard to stations.
   1, search for station
   0, add new station
   -1, update previous station (i.e., use station record provided in ESINFO(141)).
The following files are associated with the indicated logical units.

SIGMS.DIR (directory file) logical unit = 1.
SIGMXXXX.DAT (event data file, where XXXX is the event designation number) logical unit = 2.
SIGMSDIR.FRE (directory freeblock file) logical unit = 3.
SIGMXXXX.FRE (freeblock file for event data file) logical unit = 4.
NXREC1, NXREC2, NXREC3, NXREC4 are the associated variables for all possible open files.

Comments:

1) If an event is either flagged as new (i.e., not presently in the data base) or not found in a search, the station flag is set to zero to indicate that the stations are also new.

2) For the most efficient execution of this routine, the event-station information for a given event should be provided consecutively during a given run.

3) Update is written in such a way that measurements associated with an existing label can be replaced by the value in the information packet.

4) In general, the data base structure implemented by this routine is expandable with respect to events, stations, and measurements (linklist structure).

Restrictions:

1) The directory file must be open on logical unit 1 and the directory freeblock file must be open on logical unit 3.

2) Logical units 2 and 4 must be free.
3) The number of associated measurement labels and values must be ≤ 62. If the number is less than 62, the last measurement value must be followed by a word containing four blank characters.

Source File:

update.for

Associated Source Files:

search.for
chkfre.for
APPENDIX B
LISTINGS OF SIGNAL MEASUREMENT DATA BASE (SMDB) PROGRAMS

This appendix contains compiled listings of various data base management and demonstration programs. They are COPY99, INITIAL, DDBASE, EVLIST, NAVE, and RETURN. A modified version of the main routine NAVE is also included as NAVE2.
INITIAL

UNIX fortran iv vs1-11 source listing

2001 this routine initializes the signal measurement data base
2002 real*4 evbuf(32), rnam1(4), rnam2(4)
2003 integer*4 ndsg, nev, inte, ndr, mndr, nfb
2004 logical*1 temp1(4), temp2(4)
2005 equivalence (evbuf(1), ndsg), (evbuf(30), nev), (rnam1(2), temp1)
2006 equivalence (rnam2(2), temp2)
2007 data rnam1/'sigm', 'dat', rnam2/'sigm', 'fre', '/
2008 c call setfil (1, 'sigms.dir')
2009 define file 1 (32000, 128, u, nxrec1)
2010 call setfil (3, 'sigmsdir.fre')
2011 define file 3 (126, 1024, u, nxrec3)
2012 read (1, 1, end=188) nev, inte, ndr, mndr
2013 if (nev.eq.0) go to 188
2014 c delete event data files and freeblock files
2015 nev=nte
2016 do 18 k=1, nev
2017 read (1, nxev) evbuf
2018 encode (4, 12, temp1) ndsg
2019 12 format (14)
2020 encode (4, 14, temp1) temp1
2021 encode (4, 14, temp1) temp1
2022 14 format (411)
2023 call setfil (2, rnam1, 64)
2024 define file 2 (32000, 256, u, nxrec2)
2025 endfile 2
2026 encode (4, 12, temp2) ndsg
2027 encode (4, 14, temp2) temp2
2028 encode (4, 14, temp2) temp2
2029 call setfil (4, rnam2, 64)
2030 define file 4 (126, 1024, u, nxrec4)
2031 endfile 4
2032 18 continue
2033 188 continue
2034 c initialize pointers
2035 nev=3
2036 inte=0
2037 ndr=1
2038 mndr=32000
2039 write (1, 1) nev, inte, ndr, mndr
2040 write (3, 1) nfb
2041 endfile 1
2042 endfile 3
2043 stop
2044 end
this routine accesses the 9-track signal measurement tape
and does the necessary conversions and reformatting before
supplying the information to subroutine update
all info. on a tape for an event should be consecutive

real*4 esinfo(141),mslab(62),stan(58),head(375),data(180)
integer*4 ndsg,idstat,levr,iistr,iflg,iflgst,1ndsg,levorg(5)
integer*4 lidxth
logical*1 chbuf(1500),temp
equivalence (esinfo(1),ndsg),(esinfo(2),levrorg),(esinfo(7),evalt),
1 (esinfo(8),evlon), (esinfo(9),evm1b), (esinfo(10),lidxth),
2 (esinfo(11),iflg), (esinfo(12),stat),(esinfo(13),idstat),
3 (esinfo(14),iflgst), (esinfo(15),contr),(esinfo(140),levr),
4 (esinfo(141),iistr)
equivalence (chbuf,head)
data stan/'amno', 'anto', 'boco', 'chto', 'nors', 'gumo',
1 'maio', 'lasa', 'nwa0', 'grfo', 'shio', 'tato',
2 'snzo', 'ilpa', 'alpa', 'ctao', 'zobo', 'kaao',
3 'majo', 'kono', 'bfak', 'ctao', 'chgo', 'tnak',
4 'tlo', 'eiac', 'kono', 'ogdo', 'kipo', 'alqo',
5 'zipo', 'mato', 'hnme', 'rkon', 'ksrs', 'atak',
6 'ucak', 'ncak', 'njak', 'cs01', 'cs02',
7 'cs03', 'cs04', 'cs05', 'cs06', 'cs07', 'cs08',
8 '2x', '/', 'contr', 'ensc', '/ ', lndsg/*
data nmeas/25/, mslab/'ms01', 'ms02', 'ms03', 'ms04', 'ms05',
1 'ms06', 'ms07', 'ms08', 'ms09', 'ms10', 'ms11',
2 'ms12', 'ms13', 'ms14', 'ms15', 'ms16', 'ms17',
3 'ms18', 'ms19', 'ms20', 'ms21', 'ms22', 'ms23',
4 'ms24', 'ms25', '/', blank/'
data eof/'eof0'/
common /arg/, esinfo
common /asvar/ nxrec1,nxrec2,nxrec3,nxrec4
open directory and directory freeblock
call setfil (1,'signs.dir')
call setfil (1,32608,128,uxrec3)
call setfil (3,'signsdir.fre')
call setfil (2,126,1024,uxrec3)
5 continue
read in header
lunit=0
lenh=1500
lend=400
set all state flags
istate=259
call intape (lunit,head,lenh,lsw,istate)
if (lsw.ne.0) go to 995
if (lenh.ne.00) go to 7
995 do 8 k=1,4
call etoa (chbuf(k),temp)
chbuf(k)=temp
8 continue

ENSICO, INC.  B-5
UNIX fortran iv v01-11 source listing

if (head(1).eq.,eo1) go to 995
7 continue

if ebcdeic to ascii

do 10 k=29,1500

1 call etoa (chbuf(k),temp)
2 chbuf(k)=temp
3 10 continue

decode site number and use to obtain station name

decode (4,155,head) idstat

stat=statn(idstat)

decode designation number and check to see if it is a new event

decode (4,155,head) ndsg

check to see if event is the same as the last event

if (ndsg.eq.,insdf) go to 50

1nsdg=ndsg

iflgst=1

iflgst=1

decode origin time, latitude, longitude, mb, and depth

decode (29,205,head) levorg

200 format (47(4x),514)

decode (8,355,head) evlat

350 format (53(4x),f8.3)

decode (8,355,head) evlon

350 format (53(4x),f8.3)

decode (4,375,head) idpth

375 format (57(4x),14)

decode (4,400,head) evmb

400 format (58(4x),f4.2)

go to 500

50 continue

iflgst=1

note: if event was not in data base prior to this run, iflgst is changed to 0 in subroutine update since all stations will be new

read in data and convert from ibm 360/44 to pdp 11/70 floating point

2 call intape (lunit, data, lend, isw, istate)

2 if (isw.ne.,1) go to 995

2 do 60 k=1,nmeas

2 call f360f (data(k),pdptm)

2 data(k)=pdptm

2 60 continue

load measurements and labels (max. of 62) into argument list

do 70 k=1,nmeas

2 iflab=14+k*2

2 esinfo(iflab)=mslab(k)

2 ifms=1lab-1

2 esinfo(ifms)=data(k)

2 70 continue

insert blank after last measurement

ENSCO, INC. \* B-6
UNIX fortran iv v81-11 source listing

8873 if (nmoas.ne.52) esinfo(1lab+2)=blank

500 continue
8875 call update (ierr)
8877 if (ierr.eq.1) go to 995
8879 go to 5

995 endfile 1

endfile 3

stop
end
subroutine update (ierr)
c this routine establishes the signal measurement data base
c esinfo contains pertinent event-station info and flags
   iflgev=1 search for event
   iflgev=0 add event
   iflgev=-1 update previous event
   iflgst=1 search for station
   iflgst=0 add station
   iflgst=-1 update previous station
   ierr=0 no errors
   ierr=1 error in establishing data base

   logical unit=1 directory file
   =2 event data file
   =3 freeblock file for directory
   =4 freeblock file for event data file

real*4 esinfo(141), meas(2,62), key(4), dummy(20), dumy(15)
real*4 dum(9), evbuf(32), stbuf(32), mbuf(64), rnaml(4), rnam2(4)
integer*4 nev, intev, nmd, nmd, ndar, ndar, ndsg, lstev, nxev, nstat
integer*4 intst, lsstat, ncon, lstst, nxst, levr, lstr, msrec, lstms
integer*4 nxms, nfb, ifrec, iflgst, levorg(5), idpth
logical*1 temp(4), temp2(4)
equivalence (esinfo(1), ndsg), (esinfo(2), levorg), (esinfo(7), evlat),
   (esinfo(8), evlon), (esinfo(9), evmb), (esinfo(10), idpth),
   (esinfo(11), iflgev), (esinfo(12), stat), (esinfo(13), idstat),
   (esinfo(14), iflgst), (esinfo(15), contr), (esinfo(16), meas),
   (esinfo(17), levr), (esinfo(141), lstr)
equivalence (evbuf(29), lstev), (evbuf(30), nxev), (evbuf(31), nstat),
   (evbuf(32), intst), (stbuf(18), lstst), (stbuf(19), nxst),
   (stbuf(20), ncon)
equivalence (rnaml(2), temp1), (rnam2(2), temp2)
data blank, ' ', dummy(20), ' ', dumy(15), ' ', dumy(9), ' ',
data rnaml/ ' sigm', ' ', dat/ ' ', rnam2/ ' sigm', ' ', fre/ ' .
common /arg/ esinfo
common /asvar/ nxrecl,nxrec2,nxrec3,nxrec4

c c construct event file and freeblock names and open files
code (4,6,temp1) ndsg

6 format (4)

7 format (4)

code (4,6,temp2) ndsg

code (4,7,temp2) temp2

call setfil (2, rnam1)
define file 2 (32000,256,u,nxrec2)
call setfil (4, rnam2)
define file 4 (126,1024,u,nxrec4)
c

(ierr=0)

read (11) nev, intev, nmd, nmd
if (iflgev.eq.-1) go to 150
if (iflgev.eq.1) go to 100
UNIX fortran iv v81-11 source listing

1 continue
   if (ndr.ge.mndr) go to 995
   if (nev.ne.0) read (1,intev) evbuf
   call chkfre (3,0,nfb,ifrec)
   c modify backward pointer of old event
   ndr=ndr+1
   lstev=ndr
   if (nfb.gt.0) lstev=ifrec
   if (nev.ne.0) write (1,intev) evbuf
   c reset intev and appropriate pointers in new event
   nxev=intev
   intev=lstev
   lstev=0
   nstat=0
   intst=0
   write (1,intev) ndsg,ievorg,evlat,evlon,evmb,ldth,dummy,
   1 lstev,nev,nstat,intst
   ievr=intev
   nev=nev+1
   write (1,'(1) nev,intev,ndr,mndr)
   c initialize data file
   ndar=1
   mndar=32000
   write (1,'(1) ndar,mndar
   c initialize freeblock
   nfb=0
   write (4,'(1) nfb
   c set station flag to add stations
   iflgst=0
   go to 150
100 continue
   c search for event
   key(1)=ndsg
   call search (1,key,ievr,istr,icon,msrec,msind,irsch)
   if (irsch.eq.1) go to 1
150 continue
   read (1,'iev) evbuf
   if (iflgst.eq.1) go to 250
   if (iflgst.eq.1) go to 200
200 continue
   c add new station to top of list
   if (ndr.ge.mndr) go to 995
   c check to see if there is room in the data file
   read (2,'(1) ndar,mndar
   if (ndar.ge.mndr) go to 995
   if (nstat.ne.0) read (1,'intst) stbuf
   call chkfre (3,0,nfb,ifrec)
   c modify backward pointer of old station
   ndr=ndr+1
   lstst=ndr
   if (nfb.gt.0) lstst=ifrec
   if (nstat.ne.0) write (1,'intst) stbuf
   c reset intst and appro. pointers in new station
   nxst=intst
   intst=lstst
   lstst=0

ENSCO, INC.        B-9
DDBASE

(PAGE 6 OF 11)

UNIX fortran iv v81-11 source listing

```fortran
0090  nstat=nstat+1
0091  write (1,'(evr) evbuf c
0092  count measurements
0093  do 10 k=1,62
0094  if (meas(i,k).eq.blank) go to 15
0095  10 continue
0096  nmeas=k
0097  15 continue
0098  if (k.ne.62) nmeas=k-1
0100  ncon=1
0101  call chkf (4,8,nfb,ifrec)
0102  ndar=ndar+1
0103  msrec=ndar
0104  if (nfb.gt.0) msrec=ifrec
0106  lstms=0
0107  nxms=0
0108  if (nmeas.lt.31) go to 28
0110  if (ndar.ge.mndar) go to 995
0112  call chkf (4,8,nfb,ifrec)
0113  ndar=ndar+1
0114  nxms=ndar
0115  if (nfb.gt.0) nxms=ifrec
0117  28 continue
0118  write (2,'(msrec)) ((meas(i,j),j=1,31),i=2,1,-1),lstms,nxms
0119  if (nmeas.lt.31) go to 25
0121  lstms=msrec
0122  null=0
0123  write (2,'(nxms)) ((meas(i,j),j=32,62),i=2,1,-1),lstms,null
0124  25 continue
0125  write (1,'(intst) stat,tdstat,dumy,1stst,nxst,ncon,contr,msrec,
0126  1
0127  write (1,'(nev,nev,nev,nev,elem,mdyr,mdyr
0128  write (2,'(w,1,'mdyr
0129  write 4
0130  return
0131  203 continue
0132  key(2)=stat
0133  call search (2,key,evr,istr,icon,msrec,msind,lsrch)
0134  if (lsrch.eq.1) go to 151
0136  250 continue
0137  read (1,'(lstst) stbuf
0138  key(3)=contr
0139  call search (3,key,evr,istr,icon,msrec,msind,lsrch)
0140  if (lsrch.eq.0) go to 300
0142  c add new contractor block if there is room
0144  write (6,30)
0145  30 format(' all possible contr. blocks already full')
0146  go to 997
0147  50 continue
0148  c count measurements
0149  do 60 k=1,62
0150  if (meas(i,k).eq.blank) go to 65
0151  63 continue
```

ENSCO, INC.  B-10
UNIX fortran iv v91-11 source listing

```
&nmeas=k
0152 65 continue
0154 if (k.ne.62) nmeas=k-1
0156 ncon=ncon+1
0157 read (2') ndar,mndar
0158 if (ndar.ge.mndar) go to 995
0159 call chkfrec (4,6,nfb,ifrec)
0160 ndar=ndar+1
0161 msrec=ndar
0162 if (nfb.gt.0) msrec=ifrec
0163 lstms=0
0164 nxms=0
0165 if (nmeas.lt.31) go to 70
0166 if (ndar.ge.mndar) go to 995
0167 call chkfrec (4,0,nfb,ifrec)
0168 ndar=ndar+1
0169 nxms=ndar
0170 if (nfb.gt.0) nxms=ifrec
0171 70 continue
0172 write (2',msrec) (meas(i,j),j=1,31),i=2,1,-1),lstms,nxms
0173 if (nmeas.lt.31) go to 75
0174 lstms=msrec
0175 null=0
0176 write (2',nxms) (meas(i,j),j=32,62),i=2,1,-1),lstms,null
0177 75 continue
0178 c store contractor info
0179 ic=21+3*(ncon-1)
0180 stbuf(ic)=constr
0181 stbuf(ic+1)=msrec
0182 stbuf(ic+2)=nmeas
0183 write (1',ist) stbuf
0184 write (1',1) nev,ntev,ndr,mndar
0185 write (2',1) ndar,mndar
0186 endfile 2
0187 endfile 4
0188 return
0189 300 continue
0190 c loop over measurements---searching and updating
0191 do 98 k=1,62
0192 if (meas(1,k).eq.blank) go to 999
0193 key(4)=meas(1,k)
0194 call search (4,key,levr,istr,icon,msrec,msind,irsch)
0195 if (irsch.eq.1) go to 88
0196 read (2'msrec) msbuf
0197 write (2'msrec) msbuf
0198 go to 98
0199 98 continue
0200 c when measurement not found, msrec is last meas. record
0201 msind=mod(nmeas,31)+1
0202 read (2'msrec) msbuf
0203 if (msind.le.31) go to 85
0204 if (ndar.ge.mndar) go to 995
0205 msind=1
0206 call chkfrec (4,0,nfb,ifrec)
0207 ndar=ndar+1
0208 nxms=ndar
```

ENSICO, INC.

B-11
UNIX fortran iv v81-11 source listing

if (nfb.gt.0) nxms=ifrec
msbuf(64)=nxms
write (2,msrec) msbuf
do 84 n=1,64
msbuf(n)=s.0
msbuf(63)=msrec
msrec=nxms
continue
msbuf(msind)=meas(2,k)
msbuf(msind+31)=meas(1,k)
write (2,msrec) msbuf
nmeas=nmeas+1
stbuf(icon+2)=nmeas
write (1,'(A)') stbuf
continue
995 continue
go to 999
write (6,999)
999 format (' files are full')
997 ierr=1
999 endfile 2
endfile 4
return
end
subroutine search (iflag, key, evrec, strec, icon, msrec, msind, isrch)
    c
    this routine searches for a specific keyword and returns the
    appropriate record and/or index
    c
    iflag=0   full search (needs all keys)
    iflag=1   search for event
    iflag=2   search for station (needs evrec)
    iflag=3   search for contractor (needs strec)
    iflag=4   search for measurement (needs icon)
    c
    (returns rec # in 'sigmxxxx.dat' and index into rec)
    c
    key(1)=event designation
    key(2)=station name
    key(3)=contractor name
    c
    key(4)=measurement mnemonic
    c
    isrch=0   successful search
    c
    isrch=1   unsuccessful search
    c
    integer*4 nev, intev, ndr, mdr, evrec, strec, msrec, nmeas
    integer*4 ndsg, nxev, nstat, intst, nxst, nxms, icontr(12)
    real*4 key(4), buff(32), msbuff(64)

0085 equivalence (buff(1), ndsg, stat), (buff(19), nxst), (buff(30), nxev)
0086 equivalence (buff(31), intst), (buff(32), intst), (msbuff(64), nxms),
0087                 (buff(21), icontr)
0088 common /asvar/ nxrec1, nxrec2, nxrec3, nxrec4
0089 isrch=0
0090 if (iflag.eq.0) go to 100
0091 go to (100, 200, 300, 400), iflag
0092 100 read (1,1) nev, intev, ndr, mdr
0093 if (nev.eq.0) go to 995
0094 c
0095 search through events
0096 nxev=intev
0097 do 10 k=1,nev
0098    evrec=nxev
0099    read (1,nxev) buff
0100 if (key(1).eq.float(ndsg)) go to 15
0101 10 continue
0102 write (6,14) key(1)
0103 format(14) event: ',f4.0,' not found')
0104 isrch=1
0105 go to 999
0106 15 continue
0107 if (iflag.eq.0) go to 201
0108 return
0109 200 continue
0110 c
0111 search through stations given base vector (event) record #
0112 if (evrec.eq.0) go to 995
0113 read (1, evrec) buff
0114 if (nstat.eq.0) go to 995
0115 kcount=nstat
0116 nxst=intst
0117 do 20 k=1, kount
0118 strec=nxst
0119 read (1,nxst) buff
0120 if (key(2).eq.stat) go to 25
0121 20 continue
0122 write (6,24) key(2)
0123 24 format(14) station: ',a4,' not found')
UNIX fortran iv w@1-11 source listing

8846  isrch=1
8847  go to 999
8848  25 continue
8849  if (iflag.eq.8) go to 381
8850 return
8851  380 continue
8852 c
8853  search station record for contractor index
8854  if (strec.eq.8) go to 995
8855  read ('strec') buff
8856  381 continue
8857 do 30 icon=21,38,3
8858  if (key(icon).eq.buff(icon)) go to 35
8859  35 continue
8860 write (6,34) key(icon)
8861  34 format(' contractor: ',a4,' not found')
8862  isrch=1
8863  go to 999
8864  35 continue
8865  if (iflag.eq.8) go to 480
8866 return
8867  480 continue
8868  ioff=icon-28
8869  msrec=icontr(ioff+1)
8870  nmeas=icontr(ioff+2)
8871  if (nmeas.eq.8) go to 995
8872  nbk=nmeas/31+1
8873  if (mod(nmeas,31).eq.0) nbk=nbk-1
8874  nxms=msrec
8875  do 45 n=1,nbk
8876  c
8877  set msrec to current rec #
8878  msrec=nxms
8879  read (2'nxms') msbuff
8880 c
8881  search through measurement mnemonics
8882  do 48 k=32,62
8883  if (key(k).eq.msbuff(k)) go to 50
8884  45 continue
8885  45 continue
8886  write (6,46) key(k)
8887  46 format(' measurement: ',a4,' not found')
8888  isrch=1
8889  go to 999
8890  50 continue
8891  msind=k-31
8892  go to 999
8893  995 write (6,996)
8894  995 format(' error on search')
8895 stop
8896 return
8897  999 return
8898 end

ENSCO, INC.  B-14
subroutine chkfre (lunit, iflag, nfb, ifrec)
  c  this subroutine checks the appropriate free block file and
  c  returns the total of free blocks before this call and
  c  the next free record
  c  iflag=0  returns free record in ifrec
  c  iflag=1  freeblock extended with ifrec
  c
integer*4 nfb, nnfb, ifrec, freb(256)
common /asvar/ nxrec1, nxrec2, nxrec3, nxrec4
read(lunit'1) nfb
if (nfb.eq.0 and .iflag.eq.0) return
irecb= nfb/256+2
if (mod(nfb, 256).eq.0) irecb= irecb-1
read(lunit'irecb) freb
if (iflag.eq.1) go to 10
irecb = mod(nfb, 256)
if (ind.eq.0) ind=256
ifrec = freb(ind)
nfb = nfb-1
10  go to 20
20  continue
    c  find next freeblock entry
    ind=mod(ind, 256)+1
    if (ind.eq.1) irecb= irecb+1
    freb(ind)=ifrec
    nfb= nfb+1
    write (lunit'irecb) freb
20 continue
    c  modify freeblock header
    write (lunit'1) nnfb
    return
    end
this routine performs a network average for the first
'disc' discriminants for each station for a given event

REAL*4 STAT, CONTR, MLSLAB, RMEAS(100), RL1AB(100), EVBUF(32), STBUF(32)
REAL*4 KEY(4), RMX(100), DISCR(100)
INTEGER*4 NDSG, NET, IEVR, INTST, NNXST, ISTR, MSREC, NSTAT

C
C equivalence (evbuf(32), intst), (evbuf(31), nstat), (stbuf(19), nxst),
C
C DATA CONTR/'ENS'C/, RMX/0.0,0.0,
C
COMMON /FLAGS/ IFILGMS, IFILGEV
COMMON /AEVAR/ NNXST1, NNXST2, NNXST3, NNXST4
C
CALL SETFILE (1, 'SIGMS.DIR')
DEFINE FILE 1 (32000, 128, U, NNXST1)
C
C # of discriminants to average
NDISC=21
WRITE (6, 100)
100 FORMAT (' ', 'INPUT EVENT DESIGNATION #, (I4)')
READ (5, 150) NDSG
150 FORMAT (I4)
KEY(1)=NDSG
CALL SEARCH (1, KEY, IEVR, ISTR, ICON, MSREC, MSIND, ISRCH)
IF (ISRCH.EQ.1) GO TO 999
READ (1, 'IEVR') EVBUF
IFILGMS=0
IFILGEV=1
NXXST=INTST
DO 10 K=1, NSTAT
10 READ (1, 'NXST') STBUF
CALL ACCESS (NDSG, STAT, CONTR, MLSLAB, NET, RMEAS, RL1AB, IEVR)
C
LOAD MATRIX WITH MEASUREMENTS
DO 20 J=1, NDISC
IND=J+(K-1)*NDISC
RMX(IND)=RMEAS(J)
20 CONTINUE
10 CONTINUE
C
C CONVERT FROM INTEGER*4 TO INTEGER*2
NSTAT=NSTAT+1
CALL MBIAS (RMX, NSTAT, NDISC, DISCR)
WRITE (5, 200) ( RL1AB(I), DISCR(I), I=1, NDISC)
200 FORMAT (20(I4, ': ', F7.3, 'x'))
999 ENDFILE 1
STOP
END
subroutine search (iflag, key, evrec, strec, icon, msrec, msind, isrch)
  c this routine searches for a specific keyword and returns the appropriate record and/or index
  c
  c iflag=0  full search (needs all keys)
  c iflag=1  search for event
  c iflag=2  search for station (needs evrec)
  c iflag=3  search for contractor (needs strec)
  c iflag=4  search for measurement (needs icon)
  c
  c (returns rec # in 'sigmxxxx.dat' and index into rec)
  c key(1)=event designation
  c key(2)=station name
  c key(3)=contractor name
  c key(4)=measurement mnemonic
  c isrch=0  successful search
  c isrch=1  unsuccessful search

integer*4 nev, inte, ndr, mndr, evrec, strec, msrec, nmeas
integer*4 ndsg, nxev, nstat, intst, nxst, nxms, lcontr(12)
real*4 key(4), buff(32), msbuff(64)
equivalence (buff(1), ndsg, ststat), (buff(19), nxst), (buff(38), nxev)
equivalence (buff(31), nstat), (buff(32), intst), (msbuff(64), nxms),
  1 (buff(21), lcontr)
common /asvar/ nxrec1, nxrec2, nxrec3, nxrec4
isrch=0
if (iflag.eq.0) go to 100
go to (100, 200, 300, 400)
iflag
read (1,1) nev, inte, ndr, mndr
if (nev.eq.0) go to 995

search through events
nxev=nev
do 10 k=1, nev
evrec=nxev
read (1,nxev) buff
if (key(1).eq.float(ndsg)) go to 15
10 continue
write (6,14) key(1)
14 format( ' event: ', f4.8, ' not found')
isrch=1
go to 999

continue
if (iflag.eq.0) go to 201
return

continue
search through stations given base vector (event) record #
if (evrec.eq.0) go to 995
read (1, evrec) buff
201 if (nstat.eq.0) go to 995
kount=nstat
nxst=intst
do 25 k=1, kount
strec=nxst
read (1,nxst) buff
if (key(2).eq.stat) go to 25
25 continue
write (6,24) key(2)
24 format( ' station: ', a4, ' not found')
UNIX fortran iv v81-11 source listing

isrch=1

go to 999

25 continue

if (iflag.eq.0) go to 301

return

300 continue

search station record for contractor index

if (strec.eq.0) go to 995

read (1,'strec') buff

301 continue

do 30 icon=1,3,3

if (key(3).eq. buff(icon)) go to 35

30 continue

write (6,34) key(3)

34 format(' contractor: ','a4,' not found')

isrch=1

go to 999

35 continue

if (iflag.eq.0) go to 400

return

400 continue

ioff=icon-20

msrec=icontr(ioff+1)

nmeas=icontr(ioff+2)

if (nmeas.eq.0) go to 995

nblk=nmeas/31+1

if (mod(nmeas,31).eq.0) nblk=nblk-1

nxms=msrec

nblk=nblk

do 45 n=1,nblk

45 continue

msrec=nxms

c

read (2,'nxms') msbuff

c

search through measurement mnemonics

do 40 k=32,62

if (key(4).eq.msbuff(k)) go to 50

40 continue

45 continue

write (6,46) key(4)

46 format(' measurement: ','a4,' not found')

isrch=1

go to 999

50 continue

msind=k-31

59 go to 999

995 write (6,996)

996 format(' error on search')

stop

997 999 return

end
subroutine access (ndsg, stat, contr, mslab, nret, rmeas, rlab, levr)

this routine accesses the signal measurement data base

ndsg=event designation #

stat=station name (if = '****', given measurement is returned for all

stations for event)

contr=contractor name

iflgms=1 return all measurements for station (in rmeas) with assoc.

labels (in rlab)----------warning: it should be noted that

there is no guarantee all the meas. will be in the same

order across stations

iflgms=1 return only indicated measurement

mslab=label of desired measurement

nret=# of measurements returned

rmeas=measurement values

rlab=associated measurement labels

iflgev=1 search for event

iflgev=1 use previous or supplied event record # (levr)

levr=event record #

real*4 stat, contr, mslab, rmeas(1), rlab(1), evbuf(32), stbuf(32)

real*4 buff(64), key(4), rnaml(4)

integer*4 ndsg, levr, iestr, msrec, nxms, nstat, intst, nxst, icontr(12)

integer*4 nret

logical*1 templ(4)

equivalence (evbuf(31), nstat), (evbuf(32), intst), (stbuf(19), nxst),

1 (buff(64), nxms), (stbuf(21), icontr)

equivalence (rnaml(2), templ)

data rnaml/'sigm', '.', 'dat', '/', star/****/

common /flags/ iflgms, iflgev

common /asvar/ nxrec1, nxrec2, nxrec3, nxrec4

construct data file name and open file (directory file should already be

open)

encode (4,1,templ) ndsg

1 format (14)

decode (4,2,templ) templ

encode (4,2,templ) templ

2 format (41)

call setfil (2, rnaml)

define file 2 (100,256,u,nxrec2)

nret=0

key(1)=ndsg

key(3)=contr

if (iflgev.eq.0) call search (1, key, levr, iestr, icon, msrec, msind, 

1 isrch)

if (iflgev.eq.0 and .isrch.eq.1) go to 995

if (stat.eq.star) go to 500

search for specified station

key(2)=stat

call search(2, key, levr, iestr, icon, msrec, msind, isrch)

if (isrch.eq.1) go to 995

c search for contractor index

key(3)=con

call search(3, key, levr, iestr, icon, msrec, msind, isrch)

if (isrch.eq.1) go to 995

if (iflgms.eq.1) go to 30

return all measurements for station
UNIX fortran iv w01-11 source listing  

0057  read (1,'istr') stbuf
0058  ioff=icon-20
0059  msrec=iconr(ioff+1)
0060  nret=iconr(ioff+2)
0061  nb1k=nret/31+1
0062  if (mod(nret,31).eq.0) nb1k=nb1k-1
0063  nxms=msrec
0064  do 10 n=1,nb1k
0065    read (2'nxms) buff
0066      lim=nret-(n-1)*31
0067      if (lim.gt.31) lim=31
0068      do 20 k=1,lim
0069        ist=(n-1)*31+k
0070        rmeas(ist)=buff(k)
0071        rlab(ist)=buff(k+31)
0072 10 continue
0073  20 continue
0074  10 continue
0075  go to 995
0076  30 continue
0077    key(4)=mslab
0078    call search (4,key,ilevr,istr,icon,msrec,msind,lsrch)
0079    if (lsrch.eq.1) go to 995
0080    read (2'msrec) buff
0081    nret=1
0082    rmeas(nret)=buff(msind)
0083    go to 995
0084  500 continue
0085  500 continue
0086    return one measurement for all stations across event
0087    key(4)=mslab
0088    read (1'ilevr) evbuf
0089    nxst=inst
0090    do 40 k=1,nstat
0091      istr=nxst
0092      read (1'nxst) stbuf
0093      search for contractor index
0094    call search (3,key,ilevr,istr,icon,msrec,msind,lsrch)
0095    if (lsrch.eq.1) go to 40
0096    search for measurement
0097    call search (4,key,ilevr,istr,icon,msrec,msind,lsrch)
0098    if (lsrch.eq.1) go to 40
0099    read (2'msrec) buff
0100    nret=nret+1
0101    rmeas(nret)=buff(msind)
0102  40 continue
0103  995 endfile 2
0104  return
0105  end
subroutine mbias (rmx, nsta, ndisc, discr)

rmx - vector length >= nsta*ndisc
contains the discriminants (measured)
nsta - number of stations (must be <= 50)
ndisc - number of discriminants
discr - unbiased discriminant output
vector length >= ndisc

this routine performs ringdahl's maximum likelihood estimating technique for network averaging

real rmx(1), discr(1)
common / svsto / nstat, ndet(50),
xmag(50), sdnois(50)

check number of stations

if (nsta.le.50) go to 2
write (6,3)
go to 99
continue

nstat=nsta
xbias = 1000.0
sigmin = 0.2
sigmax = 1.0

do 5 i = 1, 50
sdnois(i) = 0.1
5 continue

initialize signal average

nn = 0
ns = 0
nave = 0

sigmag = 0.0

do 20 j = 1, nsta
get discriminant measurement

xmag(j) = rmx(i+(j-1)*ndisc)
if ( xmag(j) ) 30, 40, 50

noise measurement---get rid of bias

continue
nn = nn + 1
xmag(j) = xmag(j) + xbias
sigmag = sigmag + xmag(j)
nave = nave + 1
ndet(j) = 0

30 go to 20
NAVE
(PAGE 7 OF 11)

UNIX fortran iv v81-11 source listing

ignore station

continue
ndet(j) = -1
go to 28

signal measurement---get rid of bias

continue
ns = ns + 1
xmag(j) = xmag(j) - xbias
sigmag = sigmag + xmag(j)
nave = nave + 1
ndet(j) = 1
continue

starting estimate is average signal measurement

if ( nn .eq. 8 .or. ns .ge. 8 ) go to 65
smax = -1000.0
j = 1. nsta
if ( ndet(j) .ne. 0 .or. smax .gt. xmag(j) ) go to 78
smax = xmag(j)
indx = j
continue
ndet(indx) = 1
ns = 1
continue
if ( ns .ne. 8 ) go to 80
sigmag = 0.0
continue
sdsig = 0.35
sigmag = sigmag/float(nave)
call max2d (sigmag, sdsig, sigmin, sigmax)

store unbiased discriminant into output array

continue
dscr(i) = sigmag
continue
continue
return
end
subroutine max2d (x, y, sigmin, sigmax)

real likeli
real z(5)

niter = 0
nloop = 0
deltax = 0.09
deltay = 0.09

10 continue
z(1) = likeli (x, y)
z(2) = likeli (x+deltax, y)
z(3) = likeli (x-deltax, y)
yhi = y + deltax
if ( yhi .gt. sigmax ) yhi = sigmax
z(4) = likeli (x, yhi)
if ( ylo .lt. sigmin ) ylo = sigmin
ylo = y - deltax
z(5) = likeli (x, ylo)
niter = niter + 1
idir = 1
zmax = z(1)
do 20 i = 2, 5
if ( z(i) .lt. zmax ) go to 20
zmax = z(i)
idir = i
20 continue

if ( idir .eq. 1 ) go to 30
if ( idir .eq. 2 ) x = x + deltax
if ( idir .eq. 3 ) x = x - deltax
if ( idir .eq. 4 ) y = y + deltax
if ( idir .eq. 5 ) y = y - deltax
if ( y .gt. sigmax ) y = sigmax
if ( y .lt. sigmin ) y = sigmin
if ( niter .gt. 100 ) return
go to 10

30 continue
deltax = deltax/2.0
deltay = deltax/2.0
nloop = nloop + 1
if ( nloop .gt. 5 .or. niter .gt. 100 ) return
go to 10

end
real function likeli (mean, sigma)

this function returns the value of the log-likelihood function which is to be maximized w.r.t. the mean and the standard deviation

real mean

double precision ans, gauss, x

common / svsto / nsta, ndet(50),
xmag(50), sdnois(50)

x = 0.0d0

do 10 i = 1, nsta

if ( ndet(i) ) 10, 20, 30

detecting station

30 continue

arg = (xmag(i) - mean)/sigma

ans = exp (-arg*arg/2.0)/sigma

go to 40

non-detecting station

20 continue

arg = (xmag(i) - mean)/sqrt(sigma*sigma + sdnois(i)*sdnois(i))

ans = gauss (arg)

40 continue

if ( ans .lt. 1.0d-38 ) ans = 1.0d-38

x = x + dlog10 (ans)

10 continue

likeli = x

return

end
UNIX fortran iv v81-11 source listing

double precision function gauss (x)

c

double precision root2, derfc,
z

c
data root2 / 1.414213562d0 /

c
z = -x/root2

c
gauss = 0.5d0*derfc (z)

c
return

c
end
function derfc (x)
imPLICIT DOUBLE PRECISION (a-h, o-z)
derfc = 2.0d0
if ( x .lt. -10.0d0 ) return
derfc = 0.0d0
if ( x .ge. 20.0d0 ) return
derfc = 1.0d0
if ( x .eq. 0.0d0 ) return
rootpi = 1.7724538509d0
xsqr = x * x
sum = 0.0d0
xnum = 1.0d0
term = 1.0d0
if ( x .lt. 3.1d0 ) go to 10
factor = dexp (-xsqr)/(rootpi*x)
xsqr = 2.0d0 * xsqr
20 continue
sum = sum + term
xnum = xnum - 2.0d0
term1 = xnum*term/xsqr
1 if ( term1/term .lt. -1.0d0 ) go to 30
     dabs (term1) .lt. 1.0d-20 ) go to 30
     term = term1
go to 20
30 continue
derfc = factor*sum
return
10 continue
factor = dexp (-xsqr)*x*2.0d0/rootpi
xsqr = 2.0d0*xsqr
40 continue
sum = sum + term
if ( term/sum .lt. 1.0d-20 ) go to 50
xnum = xnum + 2.0d0
term = term*xsqr/xnum
go to 40
50 continue
derfc = 1.0d0 - factor*sum
return
end
UNIX fortran iv v91-11 source listing

this routine performs a network average for the first
'ndisc' discriminants for each station for a given event

real*4 stat, contr, mslab, rmeas(25), rlab(25), evbuf(32), stbuf(32)
real*4 key(4), rmx(158), discr(25), disc(25)
integer setfil
integer*4 ndsg, nret, levr, intst, nxst, istr, msrec, nstat
 equivalence (evbuf(32), intst), (evbuf(31), nstat), (stbuf(19), nxst),
 1
 (stbuf(1), stat)
data contr,'ensc',/,
common /flags/,iflgs,iflgev
common /asvar/,nxrec1,nxrec2,nxrec3,nxrec4

call setfil(1,'sigms.dir')
define file 1 (100,128,u,nxrec1)

# of discriminants to average
ndisc=21
write (6,100)
read (5,158) ndsg
100 format ('input event designation #, (i4)')
read (5,158) ndsg
158 format (14)
key(1)=ndsg
call search(1,key,1evr,istr,icon,msrec,msind,istrch)
if (istrch.eq.1) go to 999
call search(1,1evr) evbuf
iflgs=0
iflgev=1

convert integer*4 to integer*2
nsta=nstat*1
nd=3
ndav=nd
nloop=ndisc/nd+1
if (mod(ndisc,nd).eq.0) nloop=nloop-1
do 15 n=1,nloop
nxst=intst
do 15 k=1,nstat
read (1,nxst) stbuf
133 call access(nxst,stat,contr,mslab,nret,rmeas,rlab,ievrs)
load matrix with measurements
if (n.eq.nloop .and. mod(ndisc,nd).ne.0) nd=mod(ndisc,nd)
do 28 j=1,nd
nd=nd+(k-1)*nd
ioff=j+(n-1)*ndsav
rmx(iind)=rmeas(ioff)
28 continue
20 continue
18 continue
call mbias(rmx,nsta,nd,discr)
do 14 i=1,nd
ind=i+(n-1)*ndsav
discr(ind)=discr(i)
14 continue
15 continue
write (6,200)(rlab(i),discr(i),i=1,ndisc)
200 format (2005(a4,:,'f7.35x/))
999 endfile 1
99 end

UNIX fortran iv v01-11 source listing

real*4 evlat, evlon, evmb, stat, contr, rmeas(100), rlab(100)
real*4 mslab, evbuf(32), stbuf(32)
integer*4 nev, intev, ndsg, ierovg(5), idpth, intst, nstat
integer*4 nxev, nxst, ncon, lcontr(12), levr, nret, idstat
 equivalence (evbuf(1), ndsg), (evbuf(2), levorg), (evbuf(7), evlat),
    1 evbuf(8), evlon), (evbuf(9), evmb), (evbuf(10), idpth),
    2 (evbuf(33), nxev), (evbuf(31), nstat), (evbuf(32), intst)
equivalence (stbuf(1), stat), (stbuf(2), idstat), (stbuf(19), nxst),
    1 (stbuff(28), ncon), (stbuf(21), lcontr)
data contr'ensc'/
common /flags/ iflgms, iflge
common /asvar/ nxrec1, nxrec2, nxrec3, nxrec4

call setfil (1,'sigms.dir')
define file 1 (32000,128,u,nxrec1)

read (1,1) nev, intev
if (nev.eq.0) go to 999
write (6,180)

100 format ('input # to list event info; 1 to include stat. meas.',
     1 ' (11)')
read (5,150) iflg
150 format (11)
continue
write (6,280)
200 format ('input first and last event sequence #s(213)',
    1 '([<cr> to exit]')
read (5,250) ifirst, ilast
250 format (213)
if (ifirst .le. 0) go to 999
if (ifirst .gt. nev) go to 999
if (ilast .gt. nev) ilast=nev+1
loop over events
nxev=intev
do 18 k=1, ilast
write (6,260)
160 format (///)
levr=nxev
read (1,nxev) evbuf
if (k.1. i1first) go to 18
write (6,300) k, ndsg, levorg, evlat, evlon, evmb, idpth, nstat
300 format ('event seq # = ',i3,5x,'event designation # = ',i4/
  1 ' origin time : ',i2,13,2x,3(12,1x),4x,'latitude (+n) : ',
    2 f8.3,5x,'longitude (+e) : ',f8.3/' mb = ',f4.2,5x,
    3 'depth (km) = ',i4,5x,'# of stations = ',i3)
loop over stations
nxst=intst
do 320 j=1, nxst
write (6,320)
320 format(/)
read (1,nxst) stbuf
write (6,350) idstat, stat, (lcontr(j)=(j-1)*3), i=1, ncon
350 format ('station # : ',i2,10x,'station name : ',a4/
    1 ' contractors present : ',4(a4,3x))
if (iflg.eq.3) go to 20
iflgev=1

ENS CO, INC. B-28
UNIX fortran IV v01-11 source listing

if (lgms < 0)
call access (ndsg, stat, contr, mslab, nret, rmeas, rlab, levrl)
write (6,480) contr
480 format ('measurements for contractor : ', a4)
write (6,450) (rlab(i), rmeas(i), i=1, nret)
450 format (20(5(a4, ': ', f9.3,5x)/))
20 continue
10 continue
go to 1
999 endfile 1
stop
end
subroutine search (iflag,key,evrec,strec,icon,msrec,msind,tsrch)
c    this routine searches for a specific keyword and returns the
c    appropriate record and/or index

c    iflag=0  full search (needs all keys)
c    iflag=1  search for event

c    iflag=2  search for station (needs evrec)
c    iflag=3  search for contractor (needs strec)
c    iflag=4  search for measurement (needs icon)
c    key(1)=event designation

c    key(2)=station name

key(3)=contractor name

key(4)=measurement mnemonic

isrch=0  successful search

isrch=1  unsuccessful search

integer*4 nev,ntev,nrdr,mdnr,evrec,strec,msrec,nmeas
integer*4 ndsg,nev,ntst,nxst,nxms,icontr(12)
real*4 key(4),buff(32),msbuff(64)
equivalence (buff(1),ndsg,ntst), (buff(19),nxst), (buff(30),nev)
equivalence (buff(31),ntst), (buff(32),ntst), (msbuff(64),nxms),
1       (buff(21),icontr)

common/asvar/ nxrec1,nxrec2,nxrec3,nxrec4
isrch=0

if (iflag.eq.3) go to 100

go to (100,200,300,400), iflag

100 read (1,') nev,ntev,nrdr,mdnr

if (nev.eq.0) go to 995

c    search through events

nxev=ntev
do 10 k=1,nev
evrec=nev
read (1,'nxev') buff
if (key(1).eq.float(ndsg)) go to 15
10    continue
write (6,14) key(1)
14 format('event: ',a4,',' not found')
isrch=1

go to 999

15    continue
if (iflag.eq.0) go to 201
return

200    continue

c    search through stations given base vector (event) record #

if (evrec.eq.0) go to 995
read (1,'evrec') buff
201 if (ntst.eq.0) go to 995
kount=ntst
nxst=ntst
do 20 k=1,kount
strec=nxst
read (1,'nxst') buff
if (key(2).eq.stat) go to 25
20    continue
write (6,24) key(2)
24 format('station: ',a4,,' not found')

ENSOSO, INC.  B-30
EVLST
(PAGE 4 OF 6)

UNIX fortran iv v01-11 source listing

```fortran
0046       isrch=1
0047       go to 999
0048      25 continue
0049      if (iflag.eq.0) go to 301
0051      return
0052      300 continue
0053                c search station record for contractor index
0054      if (strec.eq.0) go to 995
0055      read (1,'strec') strec
0056      301 continue
0057      do 35 icon=1,30,3
0058         if (key(3).eq.buff(icon)) go to 35
0059      35 continue
0061      write (6,34) key(3)
0062      34 format(' contractor: ',a4,' not found')
0063      isrch=1
0064      go to 999
0065      35 continue
0066      if (iflag.eq.0) go to 400
0068      return
0069      400 continue
0070      ioff=icon-20
0071      msrec=icontrl(ioff+1)
0072      nmeas=icontrl(ioff+2)
0073      if (nmeas.eq.0) go to 995
0075      nbk=nmeas/31+1
0076      if (mod(nmeas,31).eq.0) nbk=nbk+1
0079      nxms=msrec
0079      do 45 n=1,nbk
0080      45 continue
0080      ioff=icon-20
0081      msrec=icontrl(ioff+1)
0082      nmeas=icontrl(ioff+2)
0083      if (nmeas.eq.0) go to 995
0085      40 continue
0086      45 continue
0087      write (6,46) key(4)
0088      46 format(' measurement: ',a4,' not found')
0089      if (iflag.eq.0) go to 999
0090      50 continue
0091      msind=k-31
0093      go to 999
0094      995 write (6,995)
0095      996 format(' error on search')
0097      stop
0098      999 return
0098      end
```
EVLIST
(PAGE 5 OF 6)

UNIX fortran iv v01-11 source listing

subroutine access (nsg, stat, contr, mslab, nret, rmeas, rlab, ifgv)
this routine accesses the signal measurement database
ndsg=event designation
stat=station name
contr=contractor name

if ifgms=0 return all measurements for station in rmeas with assoc.
labels in rlab----warning: it should be noted that
there is no guarantee all the meas. will be in the same
order across stations
if ifgms=1 return only indicated measurement
mslab=label of desired measurement
nret=# of measurements returned
rmeas=measurement values
rlab=associated measurement labels
if ifg=0 search for event
if ifg=1 use previous or supplied event record

define data file name and open file (directory file should already be
opened)
define 2 (100,256,u,nxrec2)

encode (4,1,temp1) nsg
call search(1,key,levr,istr,icon,msrec,msind,isrch)
if (ifg=eq.1) go to 995
if (ifg=eq.1 and isrch.eq.1) go to 995
if (stat.eq.star) go to 995
search for specified station
key(2)=stat
recall search(2,key,levr,istr,icon,msrec,msind,isrch)
if (isrch.eq.1) go to 995
search for contractor index
recall search(3,key,levr,istr,icon,msrec,msind,isrch)
if (isrch.eq.1) go to 995
if (ifgms.eq.1) go to 995
return all measurements for station
read(1,'istr') stbuf
ioff=icon-28
msrec=icontr(ioff+1)
nret=icontr(ioff+2)
blk=nret/31+1
if (mod(nret,31).eq.0) nblk=nblk+1
nxms=msrec
do 10 n=1,nblk
read(2'nxms') buff
lim=nret-(n-1)*31
if (lim.gt.31) lim=31
do 20 k=1,lim
ist=(n-1)*31+k
rmeas(ist)=buff(k)
rlab(ist)=buff(k+31)
20 continue
10 continue
go to 995
30 continue
key(4)=mslab
call search(4,key,ievrl,istr,icon,msrec,msind,isch)
if (isch.eq.1) go to 995
read(2'msrec') buff
nret=1
rmeas(nret)=buff(msind)
go to 995
500 continue
return one measurement for all stations across event
key(4)=mslab
read(1'ievrl') evbuf
nxst=ist
sist=nxst
read(1'nxst') stbuf
c call search for contractor index
call search(3,key,ievrl,istr,icon,msrec,msind,isch)
if (isch.eq.1) go to 40
search for measurement
call search(4,key,ievrl,istr,icon,msrec,msind,isch)
if (isch.eq.1) go to 40
read(2'msrec') buff
nret=nret+1
rmeas(nret)=buff(msind)
40 continue
995 endfile 2
return
end
RETURN
(PAGE 1 OF 5)

UNIX fortran iv v01-11 source listing

```fortran
real*4 stat, contr, mslab, rmeas(100), rlab(100)
integer*4 ndsg, nret, levr
data stat/*'**'/, mslab/*'**'/
data gumo/*'gumo'/
data /flags/ iflgms, iflgev
common /svvar/ nxrec1, nxrec2, nxrec3, nxrec4
call setfil (1,'sigms.dir')
define file 1 (32000, 120, u, nxrec1)
iflgms=1
write(6, 100)
100 format(' input event designation * (14)')
read(5, 150) ndsg
150 format(14)
write(6, 200)
200 format(' input station name (a4)')
read(5, 250) stat
250 format(a4)
if (stat.ne.stat.and. iflgms.eq.0) go to 50
write(6, 300)
300 format(' input measurement label (a4)')
read(5, 250) mslab
50 continue
contr=mslab
iflgev=0
call access (ndsg, stat, contr, mslab, nret, rmeas, rlab, levr)
write (6, 10) nret, levr
10 format(5x, 2f15)
write (6, 15) (rmeas(k), k=1, nret)
15 format(1x, f10.1)
write (6, 16) (rlab(k), k=1, nret)
16 format(1x, f10(a4, 2x))
stop
end
```

ENSICO, INC. B-34
subroutine search (iflag,key, evrec, strec, icon, msrec, msnind, isrch)
  c
  this routine searches for a specific keyword and returns the
  appropriate record and/or index
  c
  iflag=0  full search (needs all keys)
  iflag=1  search for event
  iflag=2  search for station (needs evrec)
  iflag=3  search for contractor (needs strec)
  iflag=4  search for measurement (needs icon)
  c
  (returns rec # in 'sigmxxxx.dat' and index into rec)
  c
  key(1)=event designation
  key(2)=station name
  key(3)=contractor name
  key(4)=measurement mnemonic
  c
  isrch=0  successful search
  c
  isrch=1  unsuccessful search
  c
integer*4 nev, intev, ndr, mndr, evrec, strec, msrec, nmeas
integer*4 ndsg, nxev, nstat, nxst, nxms, icontr(12)
real*4 key(4), buff(32), msbuff(64)
equivalence (buff(1), ndsg, stat1), (buff(19), nxst),
  (buff(30), nxev)
equivalence (buff(31), nstat), (buff(32), intst),
  (msbuff(64), nxms),
  (buff(21), icontr)
common /asvar/ nxrec1, nxrec2, nxrec3, nxrec4
isrch=0
if (iflag.eq.0) go to 100
if (nev.eq.0) go to 996
search through events
nev=intev
do 10 k=1, nev
  evrec=nxev
  read (1,'nxev') buff
  if (key(1).eq.float(ndsg)) go to 15
  continue
  write (6,14) key(1)
  14 format( ' event: ', f4.0, ' not found')
  isrch=1
  go to 999
15 continue
if (iflag.eq.0) go to 201
return
200 continue
search through stations given base vector (event) record #
if (evrec.eq.0) go to 996
read (1,'evrec') buff
if (nstat.eq.0) go to 996
kount=nstat
nxst=intst
do 20 k=1, kount
  strec=nxst
  read (1,'nxst') buff
  if (key(2).eq.stat) go to 25
  continue
  write (6,24) key(2)
  24 format( ' station: ', a4, ' not found')

RETURN
(PAGE 3 OF 5)

UNIX fortran iv v81-11 source listing

8046  isrch=1
8047  go to 999
8048  25 continue
8049  if (iflag.eq.0) go to 301
8050  return
8052  300 continue
8053  c search station record for contractor index
8054  if (strec.eq.0) go to 995
8055  read (1,'strec') buff
8056  301 continue
8057  do 30 icon=21,30,3
8058  if (key(3).eq.buff(icon)) go to 35
8059  30 continue
8060  write (6,34) key(3)
8061  34 format(' contractor: ',a4,' not found')
8063  isrch=1
8064  go to 999
8065  35 continue
8066  if (iflag.eq.0) go to 400
8068  return
8069  400 continue
8070  ioff=icon-20
8071  msrec=1|contr(ioff+1)
8072  nmeas=|contr(ioff+2)
8073  if (nmeas.eq.0) go to 995
8075  nbk=nmeas/31+1
8076  if (mod(nmeas,31).eq.0) nbk=nbk-1
8078  nxms=msrec
8079  do 45 n=1,nbk
8080  c set msrec to current rec #
8081  msrec=nxms
8082  read (2,'nxms') mmsb
8083  45 continue
8085  40 continue
8087  write (6,46) key(4)
8089  46 format(' measurement: ',a4,' not found')
8090  isrch=1
8091  go to 999
8092  50 continue
8093  msind=k-31
8094  go to 999
8095  995 write (6,996)
8096  996 format(' error on search')
8097  stop
8098  999 return
8099  end
RETURN
(PAGE 4 OF 5)

UNIX fortran iv w31-11 source listing page 381

subroutine access (ndsg,stat,contr,mslab,nret,rmeas,rlab,levr)
this routine accesses the signal measurement data base
ndsg=event designation #
stat=station name (if = '****', given measurement is returned for all
  stations for event)
contr=contractor name
iflgs=0 return all measurements for station (in rmeas) with assoc.
  labels (in rlab)------warning: it should be noted that
  there is no guarantee all the meas. will be in the same
  order across stations
iflgs=1 return only indicated measurement
mslab=label of desired measurement
nret=# of measurements returned
rmeas=measurement values
rlab=associated measurement labels
iflgev=0 search for event
iflgev=1 use previous or supplied event record # (levr)
levr=event record #

real*4 stat,contr,mslab,rmeas(1),rlab(1),evbuf(32),stbuf(32)
real*4 buff(64),key(4),rnaml(4)
integer*4 ndsg,levr,istr,msrec,nxms,nstat,nxst,icontr(12)
integer*4 nret
logical*1 templ(4)
equivalence (evbuf(31),nstat),(evbuf(32),intst), (stbuf(19),nxst),
1 (buff(64),nxms), (stbuf(21),icontr)
equivalence (rnaml(2),temp1)
data rnaml/'sign',
  'dat','.' star/'****'/
common /flags/ iflgs,iflgev
common /asvar/ nxrec1,nxrec2,nxrec3,nxrec4

c construct data file name and open file (directory file should already be
  open)
encode (4,1,temp1) ndsg
1 format ('i4')
decode (4,2,temp1) templ
encode (4,2,temp1) templ
2 format (4l1)
call setfil (2,rnaml)
define file 2 (100,256,u,nxrec2)

c nret=0
key(1)=ndsg
key(3)=contr
if (iflgev.eq.0) call search (1,key,levr,istr,icon,msrec,msind,
1 isrch)
if (iflgev.eq.0.and.isrch.eq.1) go to 995
if (stat.eq.star) go to 500
search for specified station
key(2)=stat
call search(2,key,levr,istr,icon,msrec,msind,isrch)
if (isrch.eq.1) go to 995
search for contractor index
call search (3,key,levr,istr,icon,msrec,msind,isrch)
if (isrch.eq.1) go to 995
if (iflgs.eq.1) go to 30
return all measurements for station
RETURN
(PAGE 5 OF 5)

UNIX fortran iv v01-11 source listing

```
0037    read (1,'istr) stbuf
0038    ioff=icon-20
0039    msrec=icontriloff+1)
0040    nret=icontriloff+2)
0041    nblk=nret/31+1
0042    if (mod(nret,31).eq.0) nblk=nblk-1
0044    nxms=msrec
0045    do 10 n=1,nblk
0046       read (2,'nxms) buff
0047       lim=nret-(n-1)*31
0048       if (lim.gt.31) lim=31
0049       do 20 k=1,lim
0051          ist=(n-1)*31+k
0052          rmeas(nret)=buff(k)
0053          rlab(nret)=buff(k+31)
0054    20  continue
0055    10  continue
0056    go to 995
0057    30  continue
0058    key(4)=mslab
0059    call search (4,key,levr,istr,icon,msrec,msind,lsrch)
0060    if (lsrch.eq.1) go to 995
0062    read (2,'msrec) buff
0063    nret=1
0064    rmeas(nret)=buff(msind)
0065    go to 995
0066    500  continue
0067    c  return one measurement for all stations across event
0068    key(4)=mslab
0069    read (1,'levr) evbuf
0070    nxst=intst
0071    do 40 k=1,nstat
0072       istr=nxst
0073    read (1,'nxst) stbuf
0074    c  search for contractor index
0075    call search (3,key,levr,istr,icon,msrec,msind,lsrch)
0076    if (lsrch.eq.1) go to 40
0077    c  search for measurement
0076    call search (4,key,levr,istr,icon,msrec,msind,lsrch)
0077    if (lsrch.eq.1) go to 40
0079    read (2,'msrec) buff
0080    nret=nret+1
0081    rmeas(nret)=buff(msind)
0082    40  continue
0083    995  endfile 2
0084    return
0085    end
```
APPENDIX C

FORMAT DESCRIPTION OF ENSCO'S RAW SIGNAL MEASUREMENT TAPE
FORMAT DESCRIPTION OF ENSCO'S RAW SIGNAL MEASUREMENT TAPE

The Raw Signal Measurement Tape is one of the tapes generated by the automated Signal Extraction Process of the Event Identification System (EIS) on the IBM 360/44 computer. This tape is a 1600 bpi nine-track tape and is written in the standard ENSCO format using physical I/O.

The standard nine-track tape format consists of a label file followed by one or more data files, followed by a trailer file. Each record in the data files represents a physical transfer of a specified number of bytes from program memory to magnetic tape. This results in efficient tape I/O and in a compact and easily interpreted tape. Figure C-1 illustrates the organization of the Raw Signal Measurement Tape. The individual components are described below.

The label file consists of two 80-byte records followed by an end-of-file mark (EOF). The first record is the IBM volume-serial header and begins with the characters VOL1. The second record is the ENSCO header and begins with the characters HDR1. The label file should be recognized by the IBM 360/44 as a standard tape label.

The data files each consist of a 1500-byte event header record followed by a 400-byte data record containing the raw signal measurements, followed by an end-of-file mark. A separate data file is generated for each event-station processed, and the data files are organized by site number (event header word 91). The stations corresponding to a given site number are presented in Table C-1 along with their locations and tectonic class.
FIGURE C-1
ORGANIZATION OF RAW SIGNAL MEASUREMENT TAPE
<table>
<thead>
<tr>
<th>Site No.</th>
<th>Station</th>
<th>Latitude</th>
<th>Longitude</th>
<th>Tectonic Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ANMO</td>
<td>34.95</td>
<td>-106.46</td>
<td>R</td>
</tr>
<tr>
<td>2</td>
<td>ANTO</td>
<td>39.87</td>
<td>32.79</td>
<td>A</td>
</tr>
<tr>
<td>3</td>
<td>BOCO</td>
<td>4.59</td>
<td>-74.04</td>
<td>A</td>
</tr>
<tr>
<td>4</td>
<td>CHTO</td>
<td>18.79</td>
<td>98.98</td>
<td>A</td>
</tr>
<tr>
<td>5</td>
<td>NORS</td>
<td>60.84</td>
<td>10.89</td>
<td>I</td>
</tr>
<tr>
<td>6</td>
<td>GUMO</td>
<td>13.59</td>
<td>144.87</td>
<td>A</td>
</tr>
<tr>
<td>7</td>
<td>MAIO</td>
<td>36.30</td>
<td>59.49</td>
<td>A</td>
</tr>
<tr>
<td>8</td>
<td>LASA</td>
<td>46.69</td>
<td>-106.22</td>
<td>I</td>
</tr>
<tr>
<td>9</td>
<td>NWAQ</td>
<td>-32.93</td>
<td>117.24</td>
<td>I</td>
</tr>
<tr>
<td>10</td>
<td>GRFO</td>
<td>49.69</td>
<td>11.22</td>
<td>I</td>
</tr>
<tr>
<td>11</td>
<td>SHIO</td>
<td>25.57</td>
<td>91.88</td>
<td>A</td>
</tr>
<tr>
<td>12</td>
<td>TATO</td>
<td>24.98</td>
<td>121.49</td>
<td>A</td>
</tr>
<tr>
<td>13</td>
<td>SN2O</td>
<td>-41.31</td>
<td>174.70</td>
<td>A</td>
</tr>
<tr>
<td>14</td>
<td>ILPA</td>
<td>35.70</td>
<td>50.61</td>
<td>A</td>
</tr>
<tr>
<td>15</td>
<td>ALPA</td>
<td>65.00</td>
<td>-147.20</td>
<td>A</td>
</tr>
<tr>
<td>16</td>
<td>CTAO (ASRO)</td>
<td>-20.09</td>
<td>146.25</td>
<td>I</td>
</tr>
<tr>
<td>17</td>
<td>ZBOO</td>
<td>-16.27</td>
<td>-68.13</td>
<td>A</td>
</tr>
<tr>
<td>18</td>
<td>KAAO</td>
<td>34.54</td>
<td>69.04</td>
<td>A</td>
</tr>
<tr>
<td>19</td>
<td>MAJO</td>
<td>36.54</td>
<td>138.21</td>
<td>A</td>
</tr>
<tr>
<td>20</td>
<td>KONO</td>
<td>59.65</td>
<td>9.60</td>
<td>I</td>
</tr>
<tr>
<td>21</td>
<td>BFAK</td>
<td>64.77</td>
<td>-146.89</td>
<td>A</td>
</tr>
<tr>
<td>22</td>
<td>CTAO (HGLP)</td>
<td>-20.09</td>
<td>146.25</td>
<td>I</td>
</tr>
<tr>
<td>23</td>
<td>CHGO</td>
<td>18.79</td>
<td>98.98</td>
<td>A</td>
</tr>
<tr>
<td>24</td>
<td>TNAK</td>
<td>62.91</td>
<td>-156.02</td>
<td>A</td>
</tr>
<tr>
<td>25</td>
<td>TLOO</td>
<td>39.86</td>
<td>-4.01</td>
<td>I</td>
</tr>
<tr>
<td>26</td>
<td>EIAO</td>
<td>29.55</td>
<td>34.95</td>
<td>R</td>
</tr>
<tr>
<td>27</td>
<td>KONO</td>
<td>59.65</td>
<td>9.60</td>
<td>I</td>
</tr>
<tr>
<td>28</td>
<td>OGDO</td>
<td>41.07</td>
<td>-74.62</td>
<td>I</td>
</tr>
<tr>
<td>29</td>
<td>KIPO</td>
<td>21.42</td>
<td>-158.02</td>
<td>A</td>
</tr>
<tr>
<td>30</td>
<td>ALQO</td>
<td>34.94</td>
<td>-106.46</td>
<td>R</td>
</tr>
<tr>
<td>31</td>
<td>ZLPO</td>
<td>-16.27</td>
<td>-68.13</td>
<td>A</td>
</tr>
<tr>
<td>32</td>
<td>MATO</td>
<td>36.54</td>
<td>138.21</td>
<td>A</td>
</tr>
<tr>
<td>33</td>
<td>HNME</td>
<td>46.16</td>
<td>-67.99</td>
<td>I</td>
</tr>
<tr>
<td>34</td>
<td>RKON</td>
<td>50.84</td>
<td>-93.67</td>
<td>I</td>
</tr>
<tr>
<td>35</td>
<td>KSRS</td>
<td>37.45</td>
<td>127.92</td>
<td>I</td>
</tr>
<tr>
<td>36</td>
<td>ATAK</td>
<td>52.88</td>
<td>173.17</td>
<td>A</td>
</tr>
<tr>
<td>37</td>
<td>UCAX</td>
<td>66.00</td>
<td>-153.72</td>
<td>A</td>
</tr>
<tr>
<td>38</td>
<td>CNAX</td>
<td>67.45</td>
<td>-144.52</td>
<td>A</td>
</tr>
<tr>
<td>39</td>
<td>NJAK</td>
<td>65.06</td>
<td>-141.85</td>
<td>A</td>
</tr>
<tr>
<td>40-50</td>
<td>Unassigned</td>
<td>--------</td>
<td>----------</td>
<td></td>
</tr>
</tbody>
</table>
Table C-2 describes all possible event header entries. In this application, not all the entries in the header record are used. Entries in the header corresponding to seismic information for specific phases should be disregarded. Information pertaining to an event or a station, or to non-seismic data (i.e., data record length, etc.) is still meaningful.

Table C-3 defines the raw signal measurements present in the data record. The amplitude measurements (words 1-21) are corrected for distance using appropriate B-factors (Sax et al., 1978). In addition, the data entries have either +1000 or -1000 added to them to indicate whether the value is a signal measurement or a noise measurement, respectively. A zero entry indicates that no measurement is available for the corresponding quantity. No regional S or Lg data were processed for this data base.

Finally, the data files are followed by a trailer file containing one 80-byte record beginning with the characters EOIO. This record is followed by at least two consecutive end-of-file marks.
<table>
<thead>
<tr>
<th>Words</th>
<th>Data Type*</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I</td>
<td>Seismogram number.</td>
</tr>
<tr>
<td>2</td>
<td>I</td>
<td>Number of components (1 or 3).</td>
</tr>
<tr>
<td>3</td>
<td>I</td>
<td>Edit length (time points).</td>
</tr>
<tr>
<td>4</td>
<td>F</td>
<td>Number of samples per second.</td>
</tr>
<tr>
<td>5</td>
<td>F</td>
<td>Edit start time (hours).</td>
</tr>
<tr>
<td>6</td>
<td>F</td>
<td>Edit start time (minutes).</td>
</tr>
<tr>
<td>7</td>
<td>F</td>
<td>Edit start time (seconds).</td>
</tr>
<tr>
<td>8-9</td>
<td>A</td>
<td>Array name.</td>
</tr>
<tr>
<td>10-12</td>
<td>A</td>
<td>Event designation.</td>
</tr>
<tr>
<td>13-14</td>
<td>A</td>
<td>Data type ('EDIT', 'GEN', or 'DISCR').</td>
</tr>
<tr>
<td>15</td>
<td>I</td>
<td>Data record length (bytes).</td>
</tr>
<tr>
<td>16</td>
<td>I</td>
<td>Number of edited sites.</td>
</tr>
<tr>
<td>17-18</td>
<td>A</td>
<td>Data partition ('SIGNAL' or 'NOISE').</td>
</tr>
<tr>
<td>19</td>
<td>A</td>
<td>Domain ('TIME' or 'FREQ').</td>
</tr>
<tr>
<td>20-21</td>
<td>A</td>
<td>Source routine.</td>
</tr>
<tr>
<td>22</td>
<td>A</td>
<td>Data orientation code ('RAW' or 'PBD').</td>
</tr>
<tr>
<td>23</td>
<td>I</td>
<td>Maximum number of channels processed.</td>
</tr>
<tr>
<td>24</td>
<td>I</td>
<td>Number of channels deleted.</td>
</tr>
<tr>
<td>25</td>
<td>F</td>
<td>Re-sample rate.</td>
</tr>
<tr>
<td>26</td>
<td>I</td>
<td>Edit start time (seconds into day).</td>
</tr>
<tr>
<td>27</td>
<td>I</td>
<td>Edit start time (year-day).</td>
</tr>
</tbody>
</table>

*A = Alphanumeric, I = Integer, F = Floating Point
### TABLE C-2

STANDARD EVENT HEADER DESCRIPTION  
(PAGE 2 OF 7)

<table>
<thead>
<tr>
<th>Words</th>
<th>Data Type*</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>F</td>
<td>Edit length (seconds).</td>
</tr>
<tr>
<td>29-33</td>
<td>B</td>
<td>Site status table (0=present, 1=absent).</td>
</tr>
<tr>
<td>34</td>
<td>I</td>
<td>Number of uncorrectable data spikes encountered during edit.</td>
</tr>
<tr>
<td>35</td>
<td>I</td>
<td>Number of data clips encountered during edit.</td>
</tr>
<tr>
<td>36</td>
<td>I</td>
<td>Filter application code.</td>
</tr>
<tr>
<td>37</td>
<td>I</td>
<td>Smoothing code.</td>
</tr>
<tr>
<td>38</td>
<td>I</td>
<td>CH equalization code.</td>
</tr>
<tr>
<td>39</td>
<td>I</td>
<td>Calibration Code.</td>
</tr>
<tr>
<td>40</td>
<td>I</td>
<td>QC procedure code.</td>
</tr>
<tr>
<td>41</td>
<td>I</td>
<td>Taper code.</td>
</tr>
<tr>
<td>42</td>
<td>F</td>
<td>Initial beam or bandpass frequency.</td>
</tr>
<tr>
<td>43</td>
<td>F</td>
<td>Final beam or bandpass frequency.</td>
</tr>
<tr>
<td>44</td>
<td>F</td>
<td>Frequency increment.</td>
</tr>
<tr>
<td>45</td>
<td>I</td>
<td>Number of frequencies.</td>
</tr>
<tr>
<td>46</td>
<td>I</td>
<td>Length of transform (time points).</td>
</tr>
<tr>
<td>47</td>
<td>I</td>
<td>Length of data transformed, or integration gate (time points).</td>
</tr>
<tr>
<td>48</td>
<td>I</td>
<td>Number of transforms stacked.</td>
</tr>
<tr>
<td>49</td>
<td>I</td>
<td>Number of edited transforms deleted from stacking.</td>
</tr>
<tr>
<td>50</td>
<td>F</td>
<td>Noise taper coefficient.</td>
</tr>
</tbody>
</table>

*A = Alphanumeric,  
I = Integer,  
F = Floating Point*
**TABLE C-2**

STANDARD EVENT HEADER DESCRIPTION
(PAGE 3 OF 7)

<table>
<thead>
<tr>
<th>Words</th>
<th>Data Type*</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>51</td>
<td>I</td>
<td>Index of signal start, relative to edit start.</td>
</tr>
<tr>
<td>52</td>
<td>I</td>
<td>Source time (seconds into day).</td>
</tr>
<tr>
<td>53</td>
<td>I</td>
<td>Source time (year-day).</td>
</tr>
<tr>
<td>54</td>
<td>A</td>
<td>Confidence of source time (PDE code).</td>
</tr>
<tr>
<td>55</td>
<td>F</td>
<td>Source latitude (± 90° N).</td>
</tr>
<tr>
<td>56</td>
<td>F</td>
<td>Source longitude (0-360° E).</td>
</tr>
<tr>
<td>57</td>
<td>F</td>
<td>Source depth (km).</td>
</tr>
<tr>
<td>58-59</td>
<td>A</td>
<td>Information source.</td>
</tr>
<tr>
<td>60</td>
<td>F</td>
<td>( m_b ).</td>
</tr>
<tr>
<td>61</td>
<td>F</td>
<td>( M_s ).</td>
</tr>
<tr>
<td>62</td>
<td>F</td>
<td>TI estimated ( M_s ).</td>
</tr>
<tr>
<td>63</td>
<td>A</td>
<td>Standard deviation of residual time.</td>
</tr>
<tr>
<td>64</td>
<td>A</td>
<td>Number of stations in PDE reporting.</td>
</tr>
<tr>
<td>65-69</td>
<td>A</td>
<td>Seismic region.</td>
</tr>
<tr>
<td>70</td>
<td>A</td>
<td>NORSAR quality.</td>
</tr>
<tr>
<td>71</td>
<td>A</td>
<td>Sub-region.</td>
</tr>
<tr>
<td>72</td>
<td>I</td>
<td>P wave arrival time (seconds into day).</td>
</tr>
<tr>
<td>73</td>
<td>I</td>
<td>P wave arrival time (year-day).</td>
</tr>
<tr>
<td>74</td>
<td>I</td>
<td>S wave arrival time (seconds into day).</td>
</tr>
<tr>
<td>75</td>
<td>I</td>
<td>S wave arrival time (year-day).</td>
</tr>
<tr>
<td>76</td>
<td>I</td>
<td>LQ wave arrival time (seconds into day).</td>
</tr>
<tr>
<td>77</td>
<td>I</td>
<td>LQ wave arrival time (year-day).</td>
</tr>
</tbody>
</table>

*A = Alphanumeric, I = Integer, F = Floating Point
<table>
<thead>
<tr>
<th>Words</th>
<th>Data Type*</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>78</td>
<td>I</td>
<td>LR wave arrival time (seconds into day).</td>
</tr>
<tr>
<td>79</td>
<td>I</td>
<td>LR wave arrival time (year-day).</td>
</tr>
<tr>
<td>80</td>
<td>I</td>
<td>Estimated LR length (seconds).</td>
</tr>
<tr>
<td>81</td>
<td>F</td>
<td>Azimuth (Great Circle beam direction).</td>
</tr>
<tr>
<td>82</td>
<td>F</td>
<td>Elevation (Great Circle beam direction).</td>
</tr>
<tr>
<td>83</td>
<td>F</td>
<td>Azimuth (Primary beam direction).</td>
</tr>
<tr>
<td>84</td>
<td>F</td>
<td>Elevation (Primary beam direction).</td>
</tr>
<tr>
<td>85</td>
<td>F</td>
<td>Source to Array Great Circle distance (degrees).</td>
</tr>
<tr>
<td>86</td>
<td>F</td>
<td>Source to Array Great Circle distance (km).</td>
</tr>
<tr>
<td>87-88</td>
<td>A</td>
<td>Seismometer type.</td>
</tr>
<tr>
<td>89</td>
<td>A</td>
<td>Recording type.</td>
</tr>
<tr>
<td>90</td>
<td>F</td>
<td>Estimated signal-to-noise ratio (dB).</td>
</tr>
<tr>
<td>91</td>
<td>I</td>
<td>Site Number.</td>
</tr>
<tr>
<td>92</td>
<td>F</td>
<td>Site latitude (± 90° N).</td>
</tr>
<tr>
<td>93</td>
<td>F</td>
<td>Site longitude (0-360° E).</td>
</tr>
<tr>
<td>94</td>
<td>I</td>
<td>Number of sub-arrays.</td>
</tr>
<tr>
<td>95-134</td>
<td>I</td>
<td>Index of the first sensor in each sub-array.</td>
</tr>
<tr>
<td>135-174</td>
<td>F</td>
<td>Reference sensor latitude (±90°N) for each sub-array.</td>
</tr>
<tr>
<td>175-214</td>
<td>F</td>
<td>Reference sensor longitude (0-360°E) for each sub-array.</td>
</tr>
<tr>
<td>215</td>
<td>A</td>
<td>Tectonic class code.</td>
</tr>
</tbody>
</table>

*A = Alphanumeric,  I = Integer,  F = Floating Point
### TABLE C-2

**STANDARD EVENT HEADER DESCRIPTION**

(PAGE 5 OF 7)

<table>
<thead>
<tr>
<th>Words</th>
<th>Data Type*</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>216</td>
<td>F</td>
<td>Long-Period</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\log_{10}$ of 50 second amplitude (vertical component).</td>
</tr>
<tr>
<td>217</td>
<td>F</td>
<td>Long-Period</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\log_{10}$ of 33 second amplitude (vertical component).</td>
</tr>
<tr>
<td>218</td>
<td>F</td>
<td>Long-Period</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\log_{10}$ of 25 second amplitude (vertical component).</td>
</tr>
<tr>
<td>219</td>
<td>F</td>
<td>Long-Period</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\log_{10}$ of 20 second amplitude (vertical component).</td>
</tr>
<tr>
<td>220</td>
<td>F</td>
<td>Long-Period</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\log_{10}$ of 17 second amplitude (vertical component).</td>
</tr>
<tr>
<td>221</td>
<td>F</td>
<td>Long-Period</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\log_{10}$ of 14 second amplitude (vertical component).</td>
</tr>
<tr>
<td>222</td>
<td>F</td>
<td>Long-Period</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\log_{10}$ of 12 second amplitude (vertical component).</td>
</tr>
<tr>
<td>223</td>
<td>F</td>
<td>Long-Period</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\log_{10}$ of 50 second amplitude (transverse component).</td>
</tr>
<tr>
<td>224</td>
<td>F</td>
<td>Long-Period</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\log_{10}$ of 33 second amplitude (transverse component).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Short-Period</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S wave, LG (surface wave, or regional P wave magnitude.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Teleseismic P wave magnitude.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Measured phase arrival time (seconds into edit)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Value of 999999 indicates no detection.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Magnitude, from first envelope peak.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mean smoothed frequency.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum mean smoothed frequency.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mean phase standard deviation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Log$_{10}$ of center frequency no. 1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Log$_{10}$ of center frequency no. 2.</td>
</tr>
</tbody>
</table>

*A = Alphanumeric, I = Integer, F = Floating Point

ENSVO, INC. C-10
<table>
<thead>
<tr>
<th>Words</th>
<th>Data Type</th>
<th>Field</th>
<th>Short-Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>225</td>
<td>F</td>
<td>Log$_{10}$ of 25 second amplitude (transverse component).</td>
<td>Log$_{10}$ of center frequency no. 5.</td>
</tr>
<tr>
<td>226</td>
<td>F</td>
<td>Log$_{10}$ of 20 second amplitude (transverse component).</td>
<td>Log$_{10}$ of center frequency no. 4.</td>
</tr>
<tr>
<td>227</td>
<td>F</td>
<td>Log$_{10}$ of 17 second amplitude (transverse component).</td>
<td>Log$_{10}$ of center frequency no. 6.</td>
</tr>
<tr>
<td>228</td>
<td>F</td>
<td>Log$_{10}$ of 14 second amplitude (transverse component).</td>
<td>Log$_{10}$ of center frequency no. 7.</td>
</tr>
<tr>
<td>229</td>
<td>F</td>
<td>Log$_{10}$ of 12 second amplitude (transverse component).</td>
<td>Log$_{10}$ of center frequency no. 8.</td>
</tr>
<tr>
<td>231-238</td>
<td>F</td>
<td>Undefined.</td>
<td>Envelope complexity.</td>
</tr>
<tr>
<td>239</td>
<td>F</td>
<td>Undefined.</td>
<td>Instantaneous frequency complexity at center frequencies no. 1-8.</td>
</tr>
<tr>
<td>240</td>
<td>F</td>
<td>Undefined.</td>
<td>m (taken from first five seconds of data.</td>
</tr>
<tr>
<td>241-248</td>
<td>F</td>
<td>Undefined.</td>
<td></td>
</tr>
<tr>
<td>249</td>
<td>F</td>
<td>Undefined.</td>
<td></td>
</tr>
</tbody>
</table>

*A = Alphanumeric, I = Integer, F = Floating Point*
<table>
<thead>
<tr>
<th>Words</th>
<th>Data Type*</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>250</td>
<td>F</td>
<td>Long-Period: Undefined.</td>
</tr>
<tr>
<td>251-265</td>
<td>F</td>
<td>Short-Period: Lg arrival time (seconds into day).</td>
</tr>
<tr>
<td>266-267</td>
<td>A</td>
<td>Variable-frequency detection ratios.</td>
</tr>
<tr>
<td>268</td>
<td>I</td>
<td>Date on which data were edited.</td>
</tr>
<tr>
<td>269</td>
<td>F</td>
<td>Start time of corresponding seismogram on a subset tape (not used for AEDS data).</td>
</tr>
<tr>
<td>270</td>
<td>F</td>
<td>B factor for P wave for source-receiver distance (AEDS data, only).</td>
</tr>
<tr>
<td>271</td>
<td>F</td>
<td>B factor for Sn wave for source-receiver distance (AEDS data, only).</td>
</tr>
<tr>
<td>272</td>
<td>F</td>
<td>B factor for lg wave for source-receiver distance (AEDS data, only).</td>
</tr>
<tr>
<td>273</td>
<td>F</td>
<td>B factor for long-period surface wave for source-receiver distance (AEDS data, only).</td>
</tr>
<tr>
<td>274</td>
<td>I</td>
<td>Edit start time index into the available data (AEDS data, only).</td>
</tr>
<tr>
<td>275-276</td>
<td></td>
<td>Available for future use.</td>
</tr>
<tr>
<td>277-336</td>
<td>I</td>
<td>Parameters used by subroutine MSDISC (AEDS data, only).</td>
</tr>
<tr>
<td>296-335</td>
<td>F</td>
<td>Sensor East Cartesian coordinates with respect to the reference sensor (km).</td>
</tr>
<tr>
<td>356-375</td>
<td>F</td>
<td>Sensor North Cartesian coordinates with respect to the reference sensor (km).</td>
</tr>
</tbody>
</table>

*A = Alphanumeric,  I = Integer,  F = Floating Point*
<table>
<thead>
<tr>
<th>Real *4 Word</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Long-Period</strong></td>
</tr>
<tr>
<td>1</td>
<td>18-22 sec Vertical Log A/T *</td>
</tr>
<tr>
<td>2</td>
<td>50 sec Vertical Log A</td>
</tr>
<tr>
<td>3</td>
<td>33.3 sec Vertical Log A</td>
</tr>
<tr>
<td>4</td>
<td>25 sec Vertical Log A</td>
</tr>
<tr>
<td>5</td>
<td>20 sec Vertical Log A</td>
</tr>
<tr>
<td>6</td>
<td>17 sec Vertical Log A</td>
</tr>
<tr>
<td>7</td>
<td>14 sec Vertical Log A</td>
</tr>
<tr>
<td>8</td>
<td>12 sec Vertical Log A</td>
</tr>
<tr>
<td>9</td>
<td>25 sec Transverse Log A</td>
</tr>
<tr>
<td></td>
<td><strong>Short-Period</strong></td>
</tr>
<tr>
<td>10</td>
<td>P-wave log A/T (Δ&lt;20°)</td>
</tr>
<tr>
<td>11</td>
<td>S-wave log A/T (Δ&lt;20°)</td>
</tr>
<tr>
<td>12</td>
<td>Lg-wave log A/T (Δ&lt;20°)</td>
</tr>
<tr>
<td>13</td>
<td>P-wave log A/T (Δ&gt;20°)</td>
</tr>
<tr>
<td>14</td>
<td>0.316 Hz Vertical Log A</td>
</tr>
<tr>
<td>15</td>
<td>0.501 Hz Vertical Log A</td>
</tr>
<tr>
<td>16</td>
<td>0.794 Hz Vertical Log A</td>
</tr>
<tr>
<td>17</td>
<td>1.259 Hz Vertical Log A</td>
</tr>
<tr>
<td>18</td>
<td>1.995 Hz Vertical Log A</td>
</tr>
<tr>
<td>19</td>
<td>3.162 Hz Vertical Log A</td>
</tr>
<tr>
<td>20</td>
<td>5.012 Hz Vertical Log A</td>
</tr>
<tr>
<td>21</td>
<td>7.943 Hz Vertical Log A</td>
</tr>
<tr>
<td>22</td>
<td>Maximum mean frequency</td>
</tr>
<tr>
<td>23</td>
<td>Mean phase standard deviation</td>
</tr>
<tr>
<td>24</td>
<td>Broadband complexity</td>
</tr>
<tr>
<td>25</td>
<td>Minimum narrowband complexity</td>
</tr>
</tbody>
</table>

* Log A/T = logarithm (base 10) amplitude/period + B-factor
** Log A = logarithm (base 10) amplitude + B-factor
REFERENCES

Sax, R. L., and Technical Staff, 1978; Event Identification - Applications to Area of Interest Events, Technical Report No. 20, Texas Instruments Report No. ALEX(01)-TR-78-08, AFTAC Contract Number F08606-77-C-0004, Texas Instruments Incorporated, Dallas, TX.