

United States
Department of the Interior
U.S. Geological Survey

Programs WSPRINT and VIEW
A Method for Creating Small ASCII waveform files
on the Eclipse Minicomputer

by Peter R. Stevenson
345 Middlefield Road
Menlo Park, California 94025

Open-File Report 81-829

This report is preliminary and has not been reviewed for conformity with U.S. Geological Survey editorial standards.

Any use of trade names is for descriptive purposes only and does not imply endorsement by USGS.

Table of Contents

	<u>page</u>
Introduction.....	3
Description of WSPRINT operation.....	4
When phase cards are absent.....	7
Description of VIEW operation.....	7
Example of WSPRINT use.....	8
Example of VIEW plot.....	9
Description of SPRINT file format.....	10
Demultiplexed seismic data files.....	14
WSPRINT program Flow.....	17
How to send SPRINT files to UNIX.....	19
How to use the Sequential I/O routines.....	22
Program listings:	
WSPRINT.....	26
SEQUENTIAL I/O.....	33
VIEW.....	45

Introduction

The program WSPRINT is designed to create small ASCII text files containing waveform data from the Central California network. It presupposes the existence of standard event data files acquired by off-line digitizing of the 1 inch analog network tapes, using the standard central California eclipse software. The intent is to provide a small subset of the standard data in a more flexible format for use in either the eclipse or other systems.

The files written in this format are referred to as SPRINT files (Seismic Printable). The data included in these files are the event ID, earthquake location, station description, P and S phase information and wave-form data.

The data within a SPRINT file is organized into variable-length logical records. Each logical record corresponds to a portion of a seismic trace. More than one record may correspond to the same trace.

View is a simple TEKTRONIX display program which serves as an adjunct to WSPRINT. Its purpose is to allow the user to view the waveform data within the SPRINT file.

Description of WSPRINT operation

In order to use WSPRINT, it is necessary to have available the three event files (-.TR, -.TD, -.EQ) corresponding to the event of interest.

These files are described in the section entitled Demultiplexed seismic data files. The program is executed by typing WSPRINT into the console. The program responds by asking
ARE YOU A NEW USER (Y=1)?

If the response is one, a brief description of the program is presented on the terminal, otherwise the program continues. The next prompt is

HOW MANY EVENTS?

After entering the number of earthquakes to be processed,
ENTER ID NUMBER:

The ID number is a 9 digit identifier assigned to the first requested earthquake by the CALNET software when the event was selected for digitizing. The next prompt is

NAME OF FILE WHICH CONTAINS STATION NAMES?

The file in question is a text file previously prepared by the user, which contains the names of those stations which are to be processed in this session. The format for each line of this file is 4 characters followed by a carriage return.

The next prompt is:

HOW MANY SPRINT RECORDS IN A STATION SET?

The idea here is to allow several sections of the trace to be obtained for each station. For example one might want a few seconds of P phase data, the S phase data, and then a larger section of coda data or perhaps the noise data preceding the P-phase.

DEFINE STATION RECORD #1

START-TIME = MARKER-TIME+OFFSET

INPUT START MARKER (P, S, C, O, D,)

The basic concept is to define each logial record as consisting of a start time and a duration time. The start time consists of a 'marker-time' and an offset (in seconds) from that marker time. The marks are P for P-phase, S for S-phase, C for coda time (origin + 30 secs), O for origin time, and D for start of digitizing.

INPUT START OFFSET

A typical example would be -1 if the marker had been P. This would start the record 1 second before the P-pick time.

DURATION = MAG*MULTIPLIER+CONSTANT

INPUT MULTIPLIER

The intent here is to allow record length to be keyed to the events' estimated magnitude. If a constant length were desired, then 0 would be an appropriate multiplier.

INPUT CONSTANT

This is the number of seconds to be added to MAG*MULTIPLIER term. If the multiplier is zero, then this becomes the duration.

DEFINE STATION RECORD #2

The above procedure is repeated n times when n is the response to the prompt for the number of sprint records per station set.

n SPRINT RECORDS PER SET

START-TIME = marker - TIME + f.fff

DURATION = MAG*g.ggg + h.hhh

START-TIME = Marker - TIME + j.jjj

etc.

OK? (Y=1)

If the SPRINT SET description is acceptable to the user, a 1 is entered, otherwise the procedure is repeated and the description is redefined.

If the response is 1, then the SPRINT file is written according to the user specifications.

The generation of SPRINT files by SISDS is quite simple. After entering the pick mode by using the PI command the appropriate trace is selected with the horizontal cross hair. After selecting the beginning of the desired time series with the vertical cross Hair, a ' $\frac{1}{4}$ ' is entered, followed by CR . The end of the time series is determined by entering a ' $\frac{1}{2}$ ' and CR after repositioning the vertical cross hair.

When phase cards are absent

In the event that a component has been digitized, but there is no phase card in the -.EQ file, alternate action is taken.

If the station component is Z, E or N, then the V component phase is read and modified by placing ** in columns 5 and 6 of the SPRINT file phase record. If the V component is also not there, then Z component is read and the same thing happens. If there are no vertical component phases, then a phase time is esimated using the origin time of the earthquake and an assumed velocity. The velocity is 6.0 km/sec if the start-marker is 'P' and 3.5 km/sec if the start-marker is 'S'. An appropriate phase card is generated, and XX is placed in columns 5 and 6.

No phase time is estimated if the start-maker is O, D, or C.

Description of VIEW operation

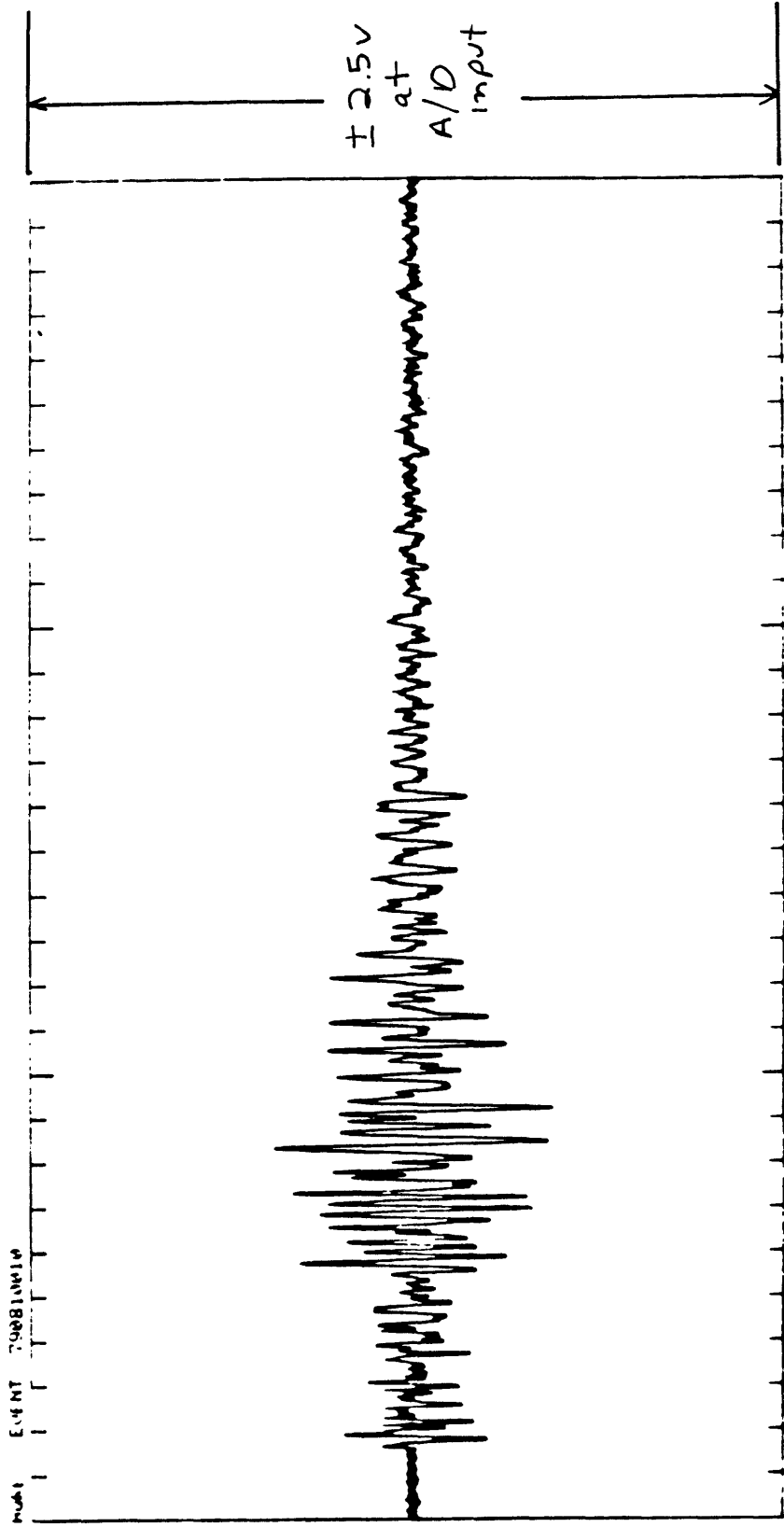
This program is intended as a visual verification of the SPRINT files created either by WSPRINT or by the (...) option of SISDS.

The input file or its link must be named 'SPRINT'. The program is invoked by typing 'VIEW' on the Tektronix 4014 terminal. The waveform portion of logical records are displayed one at a time. The computer pauses in order to allow copies to be made. To continue, the operator must enter a character followed by a carriage return. If an 'S' is entered, the program stops.

Example of WSPRINT dialogue

```
WSPRINT
CALLING TO WSPRINT.
AGE (CALL NUMBER) 1120
NEW MAIN EVENT 1
ENTER ID NUMBER
79010010
NAME OF FILE WHICH CONTAINS STATION NAMES?
NO-1A
HOW MANY SPRINT RECORDS IN A STATION SET? 2
DEFINE STATION RECORD 1
START-TIME - NUMBER-TIME + OFFSET
INPUT START NUMBER (P S C.O.D) P
INPUT START OFFSET 1
DURATION - MAGMULTIPLIER + CONSTANT
INPUT MULTIPLIER 1
INPUT CONSTANT 30
DEFINE STATION RECORD 2
START-TIME - NUMBER-TIME + OFFSET
INPUT START NUMBER (P S C.O.D) S
INPUT START OFFSET -1
DURATION - MAGMULTIPLIER + CONSTANT
INPUT MULTIPLIER 0
INPUT CONSTANT 5
2 SPRINT RECORDS PER SET
START-TIME - P-TIME + -1 000
DURATION - MAG 000 + 30 000
START-TIME - S-TIME + -1 000
DURATION - MAG 000 + 5 000
021Y111
NUMBER OF TRACES 93 TDIG 32 0000
STATION HOUR
STATION SET 1 11 7000
STATION SET 2 11 7000
ESTIMATE S-TIME FROM P-TIME
15 4960
STATION HOUR
STATION SET 1 11 7000
STATION SET 2 15 6700
STATION HOUR
STATION SET 1 11 7000
STATION SET 2 15 6700
STATION HOUR
STATION SET 1 11 7000
STATION SET 2 15 6700
END OF RUN
STOP
#
```


Example of VIEW



seconds

ECLIPSE ASCII WAVEFORM DATA
(SPRINT FORMAT)

Seismic data including event ID, Earthquake location, station information, P and S phase information and wave-form data are available in a printable form.

The data within a SPRINT file are organized into variable-length logical records. Each logical record corresponds to a seismic trace.

LOGICAL RECORD FORMAT:

- | | | |
|---------------------------|------------------------------------|------------------------------------|
| 1) Event | 'bID:b',4A2,A1,3X | 9 digit event ID |
| Date | 3I2,2X | yr, mo, day |
| Hour | I2 | |
| Minute | I2,2X | |
| Second | F8.3,2X | time of first data point |
| DT | F5.3 | sampling interval |
| | | |
| 2) Name | 'bSTATION:b', 2A2,2X | |
| Trace No. | 'TRACE#b', I3, IX | relative to .TR file |
| NPTS | I8,'bSAMPLES' | number of samples in waveform data |
| | | |
| 3) Summary Card | (40A2)* | |
| | | |
| 4) Station Card | (40A2)* | |
| | | |
| 5) P-Phase Card | (40A2)* | optional+ |
| | | |
| 6) S-Phase Card | (40A2)* | optional+ |
| | | |
| 7) SPRINT processing time | '-bSPRINTb-',1X, 3I2, 1X, 2I2, 58X | |
| | yr, mo, day, hr, min, | |
| | | |
| 8) Header terminator | '--', 78X | |
| | | |
| 9) WAVEFORM DATA | (5X, 15I5) | variable length |
| | | |
| 10) Trace Terminator | 'END-OF-TRACE', 68X | |

* Formats of the summary, station, and Phase cards depend on the Hypocenter program which produced them.

+ P and S phase cards are included only if present in the -.EQ file when the SPRINT file was generated.

HYPO 79 SUMMARY, STATION AND PHASE FORMATS

The summary card format is:

<u>item</u>	<u>columns</u>	<u>format</u>	<u>description</u>
1	1-6	3i2	year, month, day
2	7	1x	blank
3	8-11	2i2	hour, minute
4	12-17	f6.2	origin time, seconds
5	18-20	i3	latitude, degrees
6	21	a1	+ or N or - or s, if blank assume +
7	22-26	f5.2	latitude, minutes
8	27-30	i4	longitude, degrees
9	31	a1	+ or E or - or W, if blank assume +
10	32-36	f5.2	longitude, minutes
11	37	1x	blank
12	38-43	f6.2	depth, kilometers
13	44	1x	blank
14	45-50	f6.2	magnitude
15	51-53	i3	number of P phases
16	54-57	i4	gaps
17	58-60	i3	minimum distance, kilometers
18	61-62	i2	number of S phases
19	63-67	f5.2	RMS
20	68-72	f5.1	ERH
21	73-77	f5.1	ERZ
22	78	a1	"q" if an explosion or quarry
23	79	a1	quality
24	80	a1	rank of the inverse matrix for the hypo- center program (usually 4)

The station card format is:

<u>item</u>	<u>columns</u>	<u>format</u>	<u>description</u>
1	1-4	a4	name
2	5	a1	station weight
3	6	lx	blank
4	7-8	i2	latitude, degrees
5	9	a1	+ or N, - of S
6	10-14	f5.2	latitude, minutes
7	15	lx	blank
8	16-18	i3	longitude, degrees
9	19	a1	+ or E, - or W
10	20-24	f5.2	longitude, minutes
11	25-29	i5	elevation, meters
12	30-31	i2	instrument class
13	32-34	i3	attenuation (db)
14	35-37	i3	velocity model number
15	38-42	f5.2	p delay, seconds
16	43	lx	blank
17	44-45	a2	x phase (S or PN)
18	46-48	i3	x model number
19	49-53	f5.2	x delays, seconds
20	54-59	6x	blanks
21	60-65	f6.2	dt, clock correction, seconds
22	66	a1	"R" if first motion re- versed
23	67-70	a4	analog tape name, track, and vco
24	71-74	a4	remark
25	75-80	i6	yr, mo, day of the entry in the instrument history file

The phase card format is:

<u>item</u>	<u>columns</u>	<u>formats</u>	<u>description</u>
1	1-4	a4	station name
2	5	1x	blank
3	6	a1	first motion descriptor E or I
4	7-8	a2	name of the phase (e.g. P phase)
5	9	a1	first motion
6	10	i1	arrival reading weight
7	11-12	2x	blanks
8	13-14	i2	year
9	15-16	i2	month
10	17-18	i2	day
11	19	1x	blank
12	20-21	i2	hour
13	22-23	i2	minute
14	24	1x	blank
15	25-29	f5.2	second
16	30	1x	blank
17	31-35	f5.2	travel-time residual, seconds
18	36-40	f5.0	Maximum Amplitude (peak- peak)
19	41	1x	blank
20	42-45	f4.2	Period of Maximum amplitude
21	46	1x	blank
22	47-53	f6.2	epicentral distance, kilometers
23	53	1x	blank
24	54-56	13	azimuth to station, clockwise from north in degrees
25	57	1x	blank
26	58-60	i3	angle of incidence, degrees
27	61-65	f5.0	coda length in seconds
28	66	1x	blank
29	67-70	f4.1	coda magnitude
30	71-73	i3	amplitude of first half cycle in digital counts
31	74-76	i3	pulse time in digital counts (usually hund- redths of seconds)
32	77-80	2a2	remark

Demultiplexed Seismic Data Files

The demultiplexed seismic data which can be obtained by the CALNET Eclipse processing consists of three files. These files are designed -.EQ, -.TD, and -.TR, where - indicates a 9 character event ID.

The -.EQ file closely resembles the summary, station and phase cards of the HYPO71 program. The exact format of the phase records is different, however. In addition, analog tape and track assignment information is contained on the station records.

The -.TR file contains the demultiplexed seismic signals, while the -.TD file serves as a directory for the -.TR file. The Directory file contains station names and pointers to the beginning of each seismic trace.

.EQ file format:

This file is in uppercase ASCII. Each line contains exactly 80 characters and one carriage return. A character is any valid ASCII code from decimal 32 to decimal 125. The present line types implemented are as follows and occur in this order:

1. One header line
2. One summary line
3. Two blank lines
4. Multiple station lines
5. One blank line
6. Multiple phase lines.
7. Zero padding lines to fill out file.

The header line format is:

<u>item</u>	<u>columns</u>	<u>format</u>	<u>description</u>
1	1-9	9a1	unique earthquake id
2	10	1x	blank
3	11-12	a2	processing status
4	13	1x	blank
5	14-23	5a2	operator
6	24	1x	blank
7	25-30	i6	date of last change YRMODY
8	31	1x	blank

<u>item</u>	<u>columns</u>	<u>format</u>	<u>description</u>
9	32-35	i4	time of last change HRMN
10	36-39	i4	Line number of the blank line after the phase list.
11	40-43	i4	line number of the blank line preceding the phase list
12	44	1x	blank
13	45-80	18a2	header

The summary, station and phase formats are described on pp 11-13 of this report.

.TD file format:

This file is in binary, 16 bits per word for integers and 32 bits for reals. Bit arrangement is eclipse standard which is 2's complement for integers. Reals are repeated in integer form in words 61 to 66. (B TRANS Format only).

The contents of the file in words is as follows:

<u>Word</u>	<u>Description</u>
1	nothing
2-21	binary representation of 40 ascii characters
22	year of event
23	month
24	day
25	hour
26	minute
27-28	second (real)
29-30	sample interval in seconds (real)
31	number of traces
32-33	duration of digitizing in seconds (real)
34-38	binary representation of event id as 9 characters ASCII
39	digitizing software version (3, 4, 12, or 16)
40	spare
41	trace datablock index for traces 1 to 51
42	trace datablock index for traces 52 to 102
43	trace datablock index for traces 103 to 153.
.	etc.
.	etc.
60	trace datablock index for traces 970 1020
61	whole seconds of origin time
62	thousandths of seconds of origin time
63	sample interval in milliseconds
64	higher order part of duration in seconds (32767 sec.)
65	lower order part of duration in seconds (32767 sec.)
66	fractional part of duration in milliseconds
67-256	spares
information for first trace.	
257-258	station name (binary of 4 ascii characters)
259	spare
260	starting datablock number (0 is the first block in the trace file)
261	ending datablock number
262-266	same as 257-261 for second trace
267-271	same as 257-261 for third trace
.	etc.
512	zero
513-768	same as 257-512 for 52nd - 102nd trace
.	etc.

.TR file format:

This file consists of 16 bit binary integers (2's complement) where each integer represents a waveform amplitude value in digital counts at the sample rate given in the .TD file. Each individual trace for a station begins on a fileblock (256 word or 512 byte) boundary. The last fileblock is padded with zeroes after the end of the trace. Plus and minus 512 counts is full scale range from the analog-to-digital converter (10 bits). Full scale corresponds to +-2.5 volts out of the discriminators. Normal field clipping level of the VCO is +-2 volts, which corresponds to +-125 hertz modulation.

WSPRINT program flow

Open 'SPRINT'

read number of events

For each event:

 Read event ID name

read station file name

read number of records per station set

For each station set:

 Read start marker character (P, S, C, O, D)

 Read start time offset

 Read duration magnitude multiplier

 Read duration constant term

If definition not O.K.

then repeat

else continue

For each event

 Request event (RQEOK)

 open station file

 Request trace (RQTBN)

 Request P-phase and summary record (RQFAZ)

 Request S-phase and summary record (RQFAZ)

 Branch on Cases of start marker:

P: If phase-present

 then compute P-phase-digitizing time difference (TMDIF)

 Start=Difference+offset

Go to waveform processing

Else

Request Hi-gain Vertical (V). If not

present, request lo-gain Vertical (Z)

if either present

then flag phase card with '##'

start = difference + offset

go to waveform processing

else estimate P-phase time (ESTD)

compute: phase-digitizing difference (TMDIF)

start = difference + offset

Go to waveform processing

S: same as P except all references now

to S-phase instead of P-phase

C: compute origin-digitizing difference (TMDIF)

Start = origin difference + 30

go to waveform processing

O: compute origin-digitizing difference (TMDIF)

start = origin difference + offset

go to waveform processing

D: Start = offset

Waveform processing:

```

Calculate first point = start/DT + 1
Calculate number of points
write sprint header (record types 1-8)
Read waveform data (TRACD, GETTR)
Write trace terminator (record type 10)
get another event

```

End.

How to Send SPRINT files to UNIX

After WSPRINT has been run, you may wish to rename SPRINT to some other name. The reason for this might be that you wish to accumulate a number of SPRINT files, and then transfer them all at once to a tape. To do this you simply enter the CLI command `RENAME SPRINT WHATEVER1` where `WHATEVER1` will be the name you select.

When you are ready to transfer the files, you need to create a 'file of names', that is a file containing the names of the files you wish to write to a TAR tape. One way to do this is by using the XFER command. If you are seated at the Foreground (TEKTUBE) terminal, you could do the following:

```

XFER/A $ TTI: FILELIST.U
WHATEVER1
WHATEVER2
BILL
MARY
CTRL Z

```

where 'CTRL Z' means depress the control key and key Z simultaneously.

If you are on the background terminal, you would substitute \$TTI for \$TTI1 in the above example.

The next step is to mount the mag tape on drive 0. Initialize CALPROC on CALPROC2 and invoke the WRITEU program. The program asks if you want lower case conversion. As this option also substitutes Newline characters for Carriage Return characters, you should answer with a 1 (1 = yes). The program asks which filenumber to start with. The correct response is 0. The UNIX Tar program can only read the first file. The next request is for the Filename List file. This would be FILELIST.U in the example. After it has written all the files to the tape it asks the same question again. To stop, enter a carriage return.

The CLI command RELEASE MTO will rewind the tape. A sample dialogue follows.

XFER/A \$TTI FILELIST.U

SPRINT

SPEKLIST

OTHER

R

INIT DZ0.W.CALPROC

R

WRITEU

WELCOME TO TRANSF (ECLIPSE ==> UNIX)

THE LOWER CASE CONVERSION OPTION IS FOR ALL FILES THAT ONE WISHES
TO HAVE 'PRINTABLE' ON unix

THE OMISSION OF THIS OPTION WOULD BE FOR THOSE FILES THAT ONE DID NOT WANT
CHANGED IN ANY WAY (IE DIGITIZED DATA)

DO YOU WANT LOWER CASE CONVERSION? (1=YES, 0=NO)

(THIS MEANS (CARRIAGE RETURN) GOES TO (LINE FEED) ALSO) 1

FILE NUMBER TO START WITH 0

PLEASE ENTER THE FILENAME LIST FILE

FILELIST.U

SPRINT	425768. BYTES	832 BLOCKS
SPEKLIST	15702. BYTES	31 BLOCKS
OTHER	46. BYTES	1 BLOCKS

PLEASE ENTER THE FILENAME LIST FILE

STOP

R

RELEASE MT0

R

HOW TO USE THE SEQUENTIAL

I/O ROUTINES

The waveform data is organized in two random access files, the trace directory file which has a `-.TD` suffix, and the trace data file, which has a `-.TR` suffix. Earthquake information resides in the `-.EQ` file.

There are ten subroutines in this group, only five of which the user would normally call directly.

Event specific information from the directory may be retrieved by calling RQEQK. Station names can be obtained by calling RQTRC for individual traces, or RQNAM for a group of traces. Calling RQTTL retrieves the Events ASCII name.

Data is retrieved by first calling RQEQK (which calls SEQ1) once per event, then calling TRACD to initialize input for a given trace. Successive calls to GETTR will then input data to the user buffer. If it is desired to request a trace by name, a call to RQTBN will return the trace number associated with a particular station name. This number would then be used in a call to TRACD Phase, station and earthquake summary data can be retrieved by using RQFAZ.

Logical units numbers 1, 12 and 14 are assigned to `-.EQ`, `-.TR`, and `-.TD` respectively.

CALL RQEQK (LIST, EVENT)

Output	LIST(18)	event description parameters	integer
Input	EVENT(5)	event ID (9 char. ASCII)	integer

LIST(1) -

" 2	Year
" 3	Month
" 4	Day
" 5	Hour
" 6	Minute
" 7-8	Second (real)*
" 9-10	Digitizing duration (real)* in seconds
" 11-12	Sample interval (real)* in seconds
" 13-17	Event ID (9 char. ASCII)
" 18	Number of traces stored for this event

*Real variables must be equivalent to these locations, for eg. Equivalence (LIST(7), SEC), (LIST(9), DUR), (LIST(11), DELTA).

CALL RQTTL(EVTNAM)

Output	EVTNAM(20)	Event name (40 char. ASCII)	integer
--------	------------	-----------------------------	---------

CALL RQTRC(INAM, TRACE)

Output	INAM(2)	station name (4 char. ASCII)	integer
Input	Trace	requested trace number	integer

note: starting and stopping block of trace data is stored in /TRCOM/

CALL RQNAM(ISTA, NREQ, IREQ)

Output	ISTA(2, 32)	station names (4 ch. Ascii)	integer
Input	NREQ	number of requested traces	integer
Input	IREQ(32)	trace numbers of requested traces	integer

CALL SEQ1 (EVENT)

Input	EVENT(5)	EVENT ID (9 char. ASII)	integer
-------	----------	-------------------------	---------

this routine opens (EVENT) .TR and (EVENT) .TD and (EVENT) .EQ. It is called by RQEQK, so it need not be called if that routine has been called.

Logical unit numbers used are 1 for -.EQ, 12 for -.TR and 14 for -.TD

CALL TRACD(LEN, ITR, ISTART, IERR)

Input	LEN	length of user buffer (words)	integer
Input	ITR	sequence number of input trace	"
"	ISTART	first sample to be input	"
Output	IERR	error flag = 0 for normal, = 1 for error	"

CALL GETTR(LBUF, LKNT, IEND)

Output	LBUF	output buffer (retrieved trace date)	integer
"	LKNT	number of words returned in LBUF	integer
"	IEND	and flag =0 normally, =1 if last word of data trace has been read	integer

CALL RQTBN (TRNAM, ITR, NTR, IER)

Input	TRNAM (2)	station name (4 ascii char.)	integer array
"	NTR	number of traces for event	integer
Output	ITR	sequence number corresponding to TRNAM	integer
	IER	Error flag = 0 for normal, return, = 1 if station not found.	integer

CALL RQFAZ (TRNAM, IFZ, ICARD, SCARD, ISUMM, IERR)

Input	TRNAM (2)	Station name (4 ascii char.)	integer array
"	IFZ	phase name (2 ascii char.)	integer
Output	ICARD(40)	phase card image (80 char.)	integer array
"	SCARD(40)	station card image (80 char.)	integer array
"	ISUMM(40)	summary card image (80 char.)	integer array
"	IERR	error flag = -1 of error, = 0 if normal.	

Name	Function	Calls	Called by
SEQ1	Open directory trace data and earthquake data files		RQEQK
TRACD	Initiates control variables for use by GETTR	RQTRC	
GETTR	Reads data from trace data file	RDBUF	
RDBUF	Reads Disk blocks, fills buffer for GETTR	RDBLK (D.G. runtime routine)	GETTR
RQEQK	Returns event description variable to user, Reads 1st block of trace directory information	SEQ1	
RQTRC	Retrieves station name, starting & stopping block numbers stores in/TRCOM/		TRACD, RQNAM
RQNAM	Retrieves a list of station names associated with a given list of trace numbers	RQTRC	
RQTTL	Returns ASCII name of current event		
RQTBN	Returns trace sequence number for a particular station name	RQTRC	
RQFAZ	Returns phase, station, and Earthquake summary data		

ECLIPSE FORTRAN 5, VERSION 5.21 -- THURSDAY, MAY 14, 1981 2:44:01 PM

MSPRINT.FR

```

11 C THIS PROGRAM IS DESIGNED TO WRITE SPRINT FILES IN
21 C BATCH MODE ACCORDING TO THE NEW (FEB 81) FORMAT
31 C
41 C PETER R. STEVENSON 3-4-81
51 C MODIFIED 4/6/81 TO SEARCH FOR LO-Z PHASE CARD IF HI-V NOT PRESENT
61 C MODIFIED 4/8/81 TO PUT PHASE INFO INTO PHONY PHASE CARDS
71 C INTEGER A(3100),SCARD(40),SUMM(40),ICARD(40),JCARD(40)
81 C INTEGER LIST(18),IDUMM(40)
91 C EQUIVALENCE (TDIG,LIST(7)),(DELT,LIST(11)),(OTIME(1),LIST(2)),
101 C *(MTR,LIST(18))
111 C INTEGER SFILE(11),EVENT(5),OTIME(5),ATIME(5),NAM(2),
121 C *JNAM(2),OTIME(5),BTIME(5)
131 C DIMENSION ID(3,10),MARK(10),OFF(10),XMUL(10),CONS(10)
141 C INTEGER OLATD,OLONG
151 C TYPE 'WELCOME TO MPRINT!'
161 C ACCEPT 'ARE YOU A NEW USER(Y=1)?',IANS
171 C IF(IANS.NE.1)GOTO 70
181 C TYPE 'THIS PROGRAM MAKES SPRINT FILES IN BATCH MODE.'
191 C TYPE 'IT WILL PROCESS UP TO 10 EVENTS AT A TIME '
201 C TYPE 'IT WILL GENERATE SPRINT RECORDS FOR ONE OR MORE PIECES'
211 C TYPE 'OF A SEISMIC TRACE, ACCORDING TO A USER DEFINED '
221 C TYPE '"STATION SET". THIS STATION SET CONSISTS'
231 C TYPE 'OF INDIVIDUAL SPRINT RECORDS, EACH OF WHICH IS'
241 C TYPE 'DEFINED ACCORDING TO UNIQUE START AND DURATION TIMES.'
251 C TYPE 'THE START TIME IS DEFINED BY A MARKER PARAMETER'
261 C TYPE 'SUCH AS P-PHASE, OR ORIGIN TIME PLUS A CONSTANT'
271 C TYPE 'OFFSET. THE DURATION OF THE RECORD IS DEFINED'
281 C TYPE 'ACCORDING TO A MULTIPLIER TIMES THE EARTHQUAKE'
291 C TYPE 'MAGNITUDE PLUS A CONSTANT TERM.'
301 C TYPE '- - - - -'
311 C
321 70 IS = 2HS
331 IP = 2HP
341 IC = 2HC
351 IO = 2HO
361 IDIG = 2HO
371 OPEN 15,'SPRINT',ATT='A'
381 ACCEPT 'HOW MANY EVENTS?',NEV
391 DO 1 J=1,NEV
401 TYPE 'ENTER ID NUMBER!'
411 READ(11,100)(ID(JJ,J),JJ=1,5)
421 100 FORMAT(4A2,A1)
431 1 CONTINUE
441 C
451 TYPE 'NAME OF FILE WHICH CONTAINS STATION NAMES?'
461 READ(11,200)SFILE(1)
471 200 FORMAT(920)
481 C DEFINE STATION SET
491 20 ACCEPT 'HOW MANY SPRINT RECORDS IN A STATION SET?',NREC
501 IF(NREC.GT.10)TYPE 'SORRY! ONLY 10 RECORDS PER SET!'
511 IF(NREC.GT.10)NREC=10
521 DO 2 J=1,NREC
531 TYPE 'DEFINE STATION RECORD 0',J
541 TYPE 'START-TIME = MARKER-TIME + OFFSET'
551 WRITE(10,101)
561 101 FORMAT('INPUT START MARKER(P,S,C,O,D) ',Z)

```

```

57:      READ(11,201)MARK(J)
58:  201  FORMAT(A1)
59:      ACCEPT 'INPUT START OFFSET ',OFF(J)
60:      TYPE 'DURATION = MAG*MULTIPLIER + CONSTANT'
61:      ACCEPT 'INPUT MULTIPLIER ',XNUL(J)
62:      ACCEPT 'INPUT CONSTANT ',CONS(J)
63:  2    CONTINUE
64:  C
65:  C  VERIFY
66:      TYPE NREC,' SPRINT RECORDS PER SET'
67:      DO 3 J=1,NREC
68:        WRITE(10,102)MARK(J),OFF(J)
69:        WRITE(10,103)XNUL(J),CONS(J)
70:  102  FORMAT('START-TIME = ',A1,'-TIME +',F7.3)
71:  103  FORMAT('DURATION = MAG+',F7.3,' +',F7.3)
72:  C
73:  3    CONTINUE
74:      ACCEPT 'OK?(Y=1)',IY
75:      IF(IY.NE.1)GOTO 20
76:  C
77:  C  BEGIN PROCESSING EVENTS
78:      DO 30 K=1,NEV
79:      DO 31 J=1,S
80:  31    EVENT(J) = ID(J,K)
81:      CALL RREQK(LIST,EVENT)
82:  C
83:      OPEN 2,SFILE
84:      TYPE 'NUMBER OF TRACES=',NTR,' TDIG=',TDIG
85:      DO 32 J=1,NTR
86:      READ(2,300,END=33)NAM
87:  300  FORMAT(2A2)
88:      WRITE(10,1000)NAM
89:  1000 FORMAT('STATION ',2A2)
90:      CALL ROTBN(NAM,ITR,NTR,IERO)
91:      IF(IERO.NE.0)GOTO 502
92:  C
93:      IFZ = 2HP
94:      CALL RQFAZ(NAM,IFZ,ICARD,SCARD,SUMM,IER1)
95:      IFZ = 2HS
96:      CALL RQFAZ(NAM,IFZ,JCARD,SCARD,SUMM,JER1)
97:      DECODE(SUMM,151)@TIME,OSEC,OLATD,OLOND,MAG
98:  151  FORMAT(3I2,1X,2I2,F6.2,13,7X,13,14X,F6.2)
99:      IF(OLATD.GT.0.)GOTO 580
100:     DO 581 JJ=1,S
101:  581  @TIME(JJ) = @TIME(JJ)
102:       OSEC = TDIG
103:       MAG = 0
104:       TYPE 'NO SUMMARY CARD! O.T. SET TO DIG TIME.'
105:  580  DECODE(ICARD,150,ERR=570)@TIME,ASEC
106:  150  FORMAT(12X,3I2,1X,2I2,1X,F9.2)
107:       DECODE(JCARD,150,ERR=570)@TIME,BSEC
108:       GOTO 162
109:  570  TYPE '*** ERROR DECODING PHASE CARD ***'
110:  162  CONTINUE
111:  C  LOOP ON NUMBER OF SPRINT RECORDS PER TRACE
112:      DO 34 I=1,NREC
113:      TYPE 'STATION SET # ',I
114:  C
115:  C - - - - -
116:  C

```

```

117: C EACH START SYMBOL REQUIRES A DIFFERENT PATH
118: C IN ORDER TO CALCULATE THE APPROPRIATE STARTING
119: C SECOND.
120: C
121:     IF(MARK(I).EQ.IP)GOTO 51
122:     IF(MARK(I).EQ.IS)GOTO 52
123:     IF(MARK(I).EQ.IC)GOTO 53
124:     IF(MARK(I).EQ.IO)GOTO 54
125:     IF(MARK(I).EQ.IDIG)GOTO 49
126:     WRITE(10,900)MARK(I)
127: 900  FORMAT('ILLEGAL MARKER ',A1,'. D SUBSTITUTED')
128: 49   DIFF = 0.
129:     START = DIFF + OFF(I)
130:     GOTO 60
131: 51   CONTINUE
132:     IF(IER1.EQ.0)GOTO 50
133: C   GO FOR HI-GAIN
134:     JCOMP = 2H V
135:     CALL NINAM(NAM,JNAM,JCOMP)
136:     IFZ = 2HP
137:     CALL RQFAZ(JNAM,IFZ,ICARD,IDUMM,SUMM,JER2)
138:     VEL = 6.0
139:     IF(JER2.EQ.0)GOTO 50
140: C   IF NO HI-GAIN VERT GO FOR LO-GAIN VERT
141:     JCOMP = 2H Z
142:     CALL NINAM(NAM,JNAM,JCOMP)
143:     CALL RQFAZ(JNAM,IFZ,ICARD,IDUMM,SUMM,JER2)
144: C
145:     IF(JER2.NE.0)GOTO 599
146: 58   DECODE(ICARD,150)ATIME,ASEC
147: C   FLAG PHASE CARD WITH STARS
148:     ICARD(3) = 2H**
149: 50   CALL THDIF(ATIME,ASEC,DTIME,TDIG,DIFF)
150:     START = DIFF + OFF(I)
151:     GOTO 60
152: C   S-TIME
153: 52   CONTINUE
154:     IF(JER1.EQ.0)GOTO 56
155: C   NO S-PICK
156:     IF(IER1.EQ.0)GOTO 55
157: C   NO P-PICK, GO FOR HI-GAIN P
158:     JCOMP = 2H V
159:     CALL NINAM(NAM,JNAM,JCOMP)
160: C   JNAM CONTAINS VERTICAL COMPONENT NAME
161:     IFZ = 2HP
162:     CALL RQFAZ(JNAM,IFZ,ICARD,IDUMM,SUMM,JER2)
163:     VEL = 3.5
164:     IF(JER2.EQ.0)GOTO 57
165: C   IF NO HI-GAIN, GO FOR LO VERT(Z)
166:     JCOMP = 2H Z
167:     CALL NINAM(NAM,JNAM,JCOMP)
168:     CALL RQFAZ(JNAM,IFZ,ICARD,IDUMM,SUMM,JER2)
169: C   SET IFZ UP FOR S IN CASE A PHONY PHASE CARD IS NEEDED
170:     IFZ = 2HS
171:     IF(JER2.NE.0)GOTO 599
172: 57   DECODE(ICARD,150)ATIME,ASEC
173: C   FLAG PHASE CARD AS HAYRICK SUBSTITUTE
174: C   PUT ** IN COL. 5-6
175:     ICARD(3) = 2H**
176: C   ESTIMATE S-TIME FROM P-TIME

```

```

177: 55 CALL TNDIF(OTIME,ASEC,OTIME,OSEC,DIFF1)
178: VK = 1.73
179: DIFF2=DIFF1*VK
180: CALL TNDIF(OTIME,OSEC,OTIME,TDIG,ODIF)
181: START = ODIF + DIFF2 + OFF(I)
182: TYPE 'ESTIMATE S-TIME FROM P-TIME'
183: GOTO 60
184: 56 CALL TNDIF(OTIME,OSEC,OTIME,TDIG,DIFF)
185: START = DIFF + OFF(I)
186: GOTO 60
187: 599 TYPE 'UNABLE TO OBTAIN REQUESTED TIME FOR STATION'
188: CALL ESTD(SCARD,SUMM,VEL,TT,DIST)
189: XSEC = OSEC + TT
190: CALL TNDIF(OTIME,XSEC,OTIME,TDIG,XDIF)
191: START = XDIF + OFF(I)
192: TYPE 'ESTIMATE TIME FROM STATION AND EPICENTER CO-ORDS!'
193: TYPE 'PHASE VELOCITY=',VEL
194: TYPE 'TT=',TT,' DIST=',DIST,' EST. ARRIVAL TIME=',XSEC
195: C ENCODE PHONY PHASE CARD
196: ENCODE(ICARD,ISS)NAM,IFZ,OTIME,XDIF,DIST
197: 155 FORMAT(2A2,'XX',A2.4X,3I2,1X,2I2,1X,F5.2,17X,F6.2,28X)
198: GOTO 60
199: 53 CONTINUE
200: C COMPUTE CODA START TIME
201: C GET ORIGIN-DIG TIME
202: CALL TNDIF(OTIME,OSEC,OTIME,TDIG,ODIF)
203: START = ODIF + 30.
204: GOTO 60
205: 54 CONTINUE
206: C COMPUTE ORIGIN START TIME
207: CALL TNDIF(OTIME,OSEC,OTIME,TDIG,ODIF)
208: START = ODIF + OFF(I)
209: C
210: C - - - - -
211: C
212: C ALL PATHS CONVERGE HERE WITH THE STARTING
213: C SECOND HAVING BEEN CALCULATED
214: C
215: 60 IFIRST = START/DELT + 1
216: TYPE 'START-TIME=',START
217: IF(IFIRST.LE.0)IFIRST=1
218: NPTS = (MAG*KNUL(I) + CONS(I))/DELT
219: C WRITE SPRINT HEADER
220: WRITE(15,400)EVENT,OTIME,START,DELT
221: 400 FORMAT(' ID: ',5A2,2X,3I2,2X,2I2,2X,F8.3,2X,F5.3)
222: WRITE(15,401)NAM,ITR,NPTS
223: 401 FORMAT(' STATION: ',2A2,2X,' TRACE 0 ',13,1X,18,' SAMPLES')
224: WRITE(15,402)SUMM
225: 402 FORMAT(40A2)
226: WRITE(15,402)SCARD
227: WRITE(15,402)ICARD
228: IF(JER1.EQ.0)WRITE(15,402)JCARD
229: CALL FGDY(LMO,LDY,LYR)
230: CALL FGTIME(LM,LM,LSEC)
231: WRITE(15,405)LYR,LMO,LDY,LM,LM
232: 405 FORMAT('- SPRINT -',1X,3I2,1X,2I2,58X)
233: WRITE(15,406)
234: 406 FORMAT('--',78X)
235: CALL TRCD(NPTS,ITR,IFIRST,IERR)
236: IF(IERR.NE.0)GOTO 501

```

```

237:      NPTS = NPTS
238:      CALL GETTR(A,NPTS,IEND)
239:      IF(NPTS.LT.NPTS)TYPE 'SHORT DATA ERROR'
240:      WRITE(15,407)(A(JJ),JJ=1,NPTS)
241: 407    FORMAT(5X,15I5)
242:      WRITE(15,408)
243: 408    FORMAT('END-OF-TRACE',60X)
244: 34    CONTINUE
245:      GOTO 32
246: 502   TYPE 'STATION DOES NOT EXIST'
247: 32    CONTINUE
248: 33    CONTINUE
249:      REWIND 2
250: 30    CONTINUE
251:      TYPE 'END OF RUN'
252:      STOP
253: 501   TYPE 'ERROR IN READING TRACE DATA. EVENT 0',K,', TRACE 0',J
254:      STOP
255:      END

```

ECLIPSE FORTRAN 5, VERSION 5.21 -- FRIDAY, MARCH 28, 1980 10:49:51 PM

TMDIF.FR

```

1:      SUBROUTINE TMDIF(ATIME,ASEC,BTIME,BSEC,DIFF)
2:  C    DIFF=DIFFERENCE IN SECONDS BETWEEN
3:  C    ATIME(YR,MONTH,DAY,HOOR,AND MINUTE)
4:  C    ,ASEC AND BTIME,BSEC
5:  C
6:      INTEGER ATIME(5),BTIME(5)
7:  C    LOAD VARIABLES FOR CTOJL(PROTECTION)
8:      IYR=ATIME(1)
9:      IMO=ATIME(2)
10:     IDA=ATIME(3)
11:     IHR=ATIME(4)
12:     IMM=ATIME(5)
13:     ISEC=ASEC
14:     R1=ASEC-ISEC
15:  C
16:     KYR=BTIME(1)
17:     KMO=BTIME(2)
18:     KDA=BTIME(3)
19:     KHR=BTIME(4)
20:     KMM=BTIME(5)
21:     KSEC=BSEC
22:     R2=BSEC-KSEC
23:  C
24:     CALL CTOJL(IYR,JDAY1,IMO,IDA,IHR,IMM,ISEC)
25:     CALL CTOJL(KYR,JDAY2,KMO,KDA,KHR,KMM,KSEC)
26:  C
27:     DIFF=(JDAY1-JDAY2)*86400. + (IHR-KHR)*3600. +
28:     X (IMM-KMM)*60. + ISEC - KSEC +R1 - R2
29:  C
30:     RETURN
31:     END

```

ECLIPSE FORTRAN 3, VERSION 3.21 -- FRIDAY, MARCH 28, 1980 6:59:17 PM

CTOJL.FR

```

1:      SUBROUTINE CTOJL(IYR,JDAY,IMO,IDAY,IHR,ININ,ISEC)
2:      INCLUDE "HEADER.DC"
3:      C
4:      C-----
5:      C
6:      C      WRITTEN BY:      BILL ELLSWORTH
7:      C                      UNITED STATES GEOLOGICAL SURVEY
8:      C                      MENLO PARK, CALIFORNIA
9:      C
10:     C      DATE:          29 OCTOBER 1979
11:     C
12:     C      CONVERTED
13:     C          T O          JEFF HOBSON
14:     C      ECLIPSE
15:     C
16:     C-----
17:     C
18:     C---- REPLACE CALENDAR DATE AND TIME WITH JULIAN DATE AND TIME
19:     C---- ISEC MAY LIE OUTSIDE THE RANGE FROM 0 TO 59
20:     C      CALENDAR DATE IS UPDATED AS REQUIRED
21:     C      DIMENSION JDAYS(13)
22:     C      DATA JDAYS/0,31,59,90,120,151,181,212,243,273,304,334,365/
23:     C      IF (ISEC.GT.59) GO TO 20
24:     C      10 IF (ISEC.GE.0) GO TO 30
25:     C      ISEC=ISEC+60
26:     C      ININ=ININ-1
27:     C      IF (ININ.GE.0) GO TO 10
28:     C      ININ=59
29:     C      IHR=IHR-1
30:     C      IF (IHR.GE.0) GO TO 10
31:     C      IHR=23
32:     C      IDAY=IDAY-1
33:     C      IF (IDAY.GT.0) GO TO 10
34:     C      IMO=IMO-1
35:     C      IF (IMO.EQ.0) GO TO 15
36:     C      IDAY=JDAYS(IMO+1)-JDAYS(IMO)
37:     C      IF (MOD(IYR,4).EQ.0.AND.IMO.EQ.2) IDAY=29
38:     C      GO TO 10
39:     C
40:     C      15 IMO=12
41:     C      IDAY=31
42:     C      IYR=IYR-1
43:     C      GO TO 10
44:     C
45:     C      20 IF (ISEC.LE.59) GO TO 30
46:     C      ISEC=ISEC-60
47:     C      ININ=ININ+1
48:     C      IF (ININ.LE.59) GO TO 20
49:     C      ININ=0
50:     C      IHR=IHR+1
51:     C      IF (IHR.LE.23) GO TO 20
52:     C      IHR=0
53:     C      IDAY=IDAY+1
54:     C      NDAYS=JDAYS(IMO+1)-JDAYS(IMO)
55:     C      IF (MOD(IYR,4).EQ.0.AND.IMO.EQ.2) NDAYS=29
56:     C      IF (IDAY.LE.NDAYS) GO TO 20

```

```

57:      IDAY=1
58:      IMO=IMO+1
59:      IF (IMO.LE.12) GO TO 20
60:      IMO=1
61:      IYR=IYR+1
62:      GO TO 20
63:  C
64:      30 JDAY=IDAY+JDAYS(IMO)
65:      IF (MOD(IYR,4).EQ.0.AND IMO.GT.2) JDAY=JDAY+1
66:      RETURN
67:      END

```

ECLIPSE FORTRAN 9, VERSION 9.21 -- MONDAY, APRIL 6, 1981 11:22:35 PM

ESTD.FR

```

1:      SUBROUTINE ESTD(SCARD,SUMM,VEL,TT,DIST)
2:  C  SUBROUTINE TO ESTIMATE TRAVEL TIME AND DISTANCE
3:  C  FROM HYPOCENTRAL AND STATION CO-ORDINATES
4:      INTEGER OLATD,OLOND
5:      PI = 3.1415927
6:      DECODE(SCARD,152)LATD,RLAT,LOND,RLOM
7:  152  FORMAT(6X,I2,1X,F5.2,1X,I3,1X,F5.2)
8:      DECODE(SUMM,153)OLATD,OLATH,OLOND,OLONM,DEPTH
9:  153  FORMAT(17X,I3,1X,F5.2,1X,I3,1X,F5.2,1X,F6.2)
10:     OLATH=OLATH + 60.*OLATD
11:     OLONM = OLONM + 60.*OLOND
12:     RLAT = RLAT + 60 *LATD
13:     RLOM = RLOM + 60 *LOND
14:     R7 = ((LATD + OLATD)/2.)*(PI/180.)
15:     R4 = ABS(OLONM-RLOM)*1.875*COS(R7)
16:     R2 = ABS(OLATH-RLAT)*1.85
17:     DIST = SQRT(R4**2 + R2**2)
18:     TT = SQRT(DIST**2 + DEPTH**2)/VEL
19:     RETURN
20:     END

```


ECLIPSE FORTRAN 5, VERSION 5.21 -- MONDAY, APRIL 6, 1981 1:25:11 PM

HINAM.FR

```

11      SUBROUTINE HINAM(NAM,JNAM,JCOMP)
12      DIMENSION NAM(2),JNAM(2)
13      C THIS ROUTINE REPLACES THE CHAR IN THE 4TH POSITION
14      C OF NAM WITH RIGHT JUSTIFIED CHAR IN JCOMP
15      C AND STORES THE NEW NAME IN JNAM
16      C P R STEVENSON 3-23-81
17      C MODIFIED 4/6/81 TO GENERAL CASE
18      C
19      MASK1 = 377K
20      MASK2 = 177400K
21      NCOMP = IAND(JCOMP,MASK1)
22      JNAM(1) = NAM(1)
23      JNAM(2) = IAND(NAM(2),MASK2)
24      JNAM(2) = IOR(JNAM(2),NCOMP)
25      C JNAM CONTAINS NEW COMPONENT NAME
26      RETURN
27      END

```

ECLIPSE FORTRAN 5, VERSION 5.21 -- MONDAY, AUGUST 11, 1980 12:14:11 PM

SEQ1.FR

```

11      SUBROUTINE SEQ1(EVENT)
12      C
13      C INITIALIZATION ROUTINE FOR SEQUENTIAL TRACE DATA
14      C RETREIVAL. THIS ROUTINE IS CALLED FIRST TO OPEN
15      C TRACE DATA AND DIRECTORY FILES.
16      C
17      C INCLUDE 'CSEVCOM'
18      C -----
19      C THE SEQUENTIAL ACCESS PACKAGE, WHICH
20      C INCLUDES SUBROUTINES SEQ1,TRACD,GETTR,RDBUF,RREQK,
21      C RQTRC,RQNAM AND RQTL, WAS WRITTEN BY
22      C PETER R. STEVENSON OF THE U.S.G.S.
23      C IN JANUARY 1979
24      C -----
25      C
26      C TBUF = TEMPORARY STORAGE AREA
27      C EVLST = EVENT DIRECTORY INFORMATION
28      C TRINDX = TRACE INDEX
29      C DRFILE = LOGICAL UNIT OF DIRECTORY FILE
30      C
31      C PARAMETER LSZ = 45
32      C COMMON/EVCOM/TBUF(256),EVLST(LSZ),ICURR
33      C INTEGER TBUF,EVLST,ENAM(40),TIME(7),EVENT(5),TRACE
34      C *,UNQID(4),TRINDX(5),DRFIL
35      C EQUIVALENCE (EVLST(41),TRINDX(1)),(EVLST(22),TIME(1)),
36      C *(EVLST(2),ENAM(1)),(EVLST(29),DELT),(EVLST(32),DUR)
37      C
38      C PARAMETER DRFIL = 14
39      C
40      C COMMON /TRCOM/NAM(2),IFAZ,MSTRT,MEND,MTR
41      C
42      C INCLUDE 'CSIUCOM'
43      C
44      C COMMON /IOCOM/MKNT,IFIRST,LSIZ,LL,NH,IFLAG,JFLAG,LBLK,RCNT,FCNT
45      C PARAMETER NBLK = 4, NSIZ = NBLK*256, IFILE = 12
46      C
47      C INTEGER FNAM(7),GNAM(7),QNAM(7)

```

```

39:  C
40:  C
41:      INTEGER EQFIL
42:      PARAMETER EQFIL=1
43:  C BUILD UP FILENAMES FROM EVENT ID STRING
44:      DO 2 J=1,5
45:          GNAM(J) = EVENT(J)
46:          FNAM(J) = EVENT(J)
47:      2   GNAM(J) = EVENT(J)
48:  C   EXTRACT LAST CHAR OF ID AND APPEND '.'
49:      MASK = 177400K
50:      MOVE = 2H<NUL>.
51:      GNAM(5) = IAND(GNAM(5),MASK)
52:      GNAM(5) = IOR(GNAM(5),MOVE)
53:      GNAM(5) = GNAM(5)
54:      GNAM(6) = 2HEQ
55:      FNAM(5) = GNAM(5)
56:      FNAM(6) = 2HTR

57:      GNAM(6) = 2HTD
58:      FNAM(7) = 2H<NUL><NUL>
59:      GNAM(7) = FNAM(7)
60:      GNAM(7) = FNAM(7)
61:  C
62:  C   CLOSE FILES IF OPEN
63:      CALL CLOSE(EQFIL,IERR)
64:      CALL CLOSE(DRFIL,IERR)
65:      CALL CLOSE(IFILE,IERR)
66:  C
67:  C   OPEN TRACE DIRECTORY FILE
68:      OPEN DRFIL,GNAM,LEN=512
69:  C
70:  C
71:  C   SET CURRENT BLOCK COUNT TO 1
72:      ICURR = 1
73:  C
74:  C   OPEN TRACE DATA FILE
75:      OPEN IFILE,FNAM
76:  C
77:  C   OPEN EARTHQUAKE DATA FILE
78:      OPEN EQFIL,GNAM,LEN=81
79:  C
80:  C
81:      RETURN
82:      END

```

ECLIPSE FORTRAN 5, VERSION 5.21 -- MONDAY, AUGUST 11, 1990 12:15:01 PM

TRACD.FR

```

1:      SUBROUTINE TRACD(LEN,ITR,ISTART,IERR)
2:      C
3:      C  DATA DEFINITION ROUTINE FOR INDEXED-SEQUENTIAL DATA
4:      C  ORGANIZED AS TIME SERIES
5:      C
6:      C  LEN = LENGTH OF USER BUFFER
7:      C  ITR = SEQUENCE NUMBER OF INPUT TRACE
8:      C  ISTART = 1ST DATA WORD TO INPUT RELATIVE TO BEGINNING
9:      C  OF TIME SERIES
10:     C  IERR = ERROR RETURN  ERROR = -1, 0 = NORMAL RETURN
11:     C
12:     C      INCLUDE 'C$EVCOM'
13:     C  - - - - -
14:     C  THE SEQUENTIAL ACCESS PACKAGE, WHICH
15:     C  INCLUDES SUBROUTINES SEQ1,TRACD,GETTR,RDBUF,RREQK,
16:     C  RQTRC,RQYAH AND RQTL, WAS WRITTEN BY
17:     C  PETER R. STEVENSON OF THE U.S.G.S.
18:     C  IN JANUARY 1979
19:     C  - - - - -
20:     C
21:     C
22:     C      TBUF = TEMPORARY STORAGE AREA
23:     C      EVLST = EVENT DIRECTORY INFORMATION
24:     C      TRINDX = TRACE INDEX
25:     C      DRFILE = LOGICAL UNIT OF DIRECTORY FILE
26:     C
27:     C      PARAMETER LSZ = 45
28:     C      COMMON/EVCOM/TBUF(256),EVLST(LSZ),ICURR
29:     C      INTEGER TBUF,EVLST,ENAM(40),TIME(7),EVENT(5),TRACE
30:     C      *.UNQID(4),TRINDX(9),DRFIL
31:     C      EQUIVALENCE (EVLST(41),TRINDX(1)),(EVLST(22),TIME(1)),
32:     C      *(EVLST(2),ENAM(1)),(EVLST(29),DELT),(EVLST(32),DUR)
33:     C
34:     C      PARAMETER DRFIL = 14
35:     C
36:     C      INCLUDE 'C$IOCOM'
37:     C
38:     C      COMMON /IOCOM/NKNT,IFIRST,LSIZ,LL,MH,IFLAG,JFLAG,LBLK,RCNT,FCNT
39:     C      PARAMETER NBLK = 4, NSIZ = NBLK*256, IFILE = 12
40:     C
41:     C
42:     C      DIMENSION INAM(2)
43:     C
44:     C      COMMON/TRCOM/INAM(2),IFAZ,NSTRT,NEND,MTR
45:     C  RETRIEVE TRACE DIRECTORY INFO FOR TRCOM
46:     C  CALL RQTRC(INAM,ITR)
47:     C
48:     C      IERR=0
49:     C      IF(ITR.NE.MTR)IERR=-1
50:     C  ASSIGN VALUES TO IOCOM
51:     C      LSIZ = LEN
52:     C      IFIRST = MOD(ISTART,NSIZ)
53:     C  SET INPUT WORD COUNTER
54:     C      NKNT = IFIRST - 1
55:     C  ZERO INPUT BUFFER COUNTER
56:     C      LL = 0

```

```
57: C SET USER BUFFER COUNTER
58:   NM = (ISTART - IFIRST)/NSIZ
59: C SET BUFFER FLAG TO EMPTY
60:   IFLAG = 0
61: C CLEAR END OF BLOCK FLAG AND
62: C LOAD NEW LAST-BLOCK VALUE
63:   JFLAG = 0
64: C COMPUTE NUMBER OF FIRST DATA BLOCK TO PLOT
65:   NFB = (ISTART-1)/NSIZ + 1
66: C
67:   LBLK = NEND - NSTRT - NFB + 1
68: C
69: C FIX BY P.R.S. 9-7-79 TO PREVENT ACCESSING
70: C ZERO-FILL NUMBERS AT END OF DATA
71: C SET UP TOTAL DATA LENGTH CHECKING PARAMETERS
72: C RCNT = TOTAL CUMULATIVE POINTS
73: C FCNT = TOTAL VALID POINTS
74:   RCNT=0.
75:   FCNT=DUR/DELT - ISTART +1
76: C
77: C CHECK FOR BAD PARAMETERS
78:   IF(MINO(LSIZ,IFIRST).LE.0)IERR=-1
79:   RETURN
80:   END
```

ECLIPSE FORTRAN 3, VERSION 3.21 -- MONDAY, AUGUST 11, 1980 12:15:48 PM

GETTR.FR

```

1:      SUBROUTINE GETTR(LBUF,LKNT,IEND)
2:      C
3:      C INPUT ROUTINE FOR INDEXED-SEQUENTIAL DATA
4:      C ORGANIZED AS TIME SERIES
5:      C
6:      C LBUF = OUTPUT BUFFER
7:      C LKNT = NUMBER OF DATA WORDS RETURNED IN LBUF
8:      C IEND = 1 FOR EOF, 0 FOR NO EOF
9:      C
10:     C MODIFIED 9-7-79 BY P.R.S. TO RETURN DATA ONLY
11:     C THROUGH LAST VALID DATA POINT (I.E. DOES NOT
12:     C NOW RETURN ZERO-FILL NUMBERS AT END OF DATA
13:     C
14:     COMMON /TRCOM/NAH(2),IFAZ,NSTRT,NEND,NTR
15:     C
16:     INCLUDE 'C$EVCOM'
17:     C - - - - -
18:     C THE SEQUENTIAL ACCESS PACKAGE, WHICH
19:     C INCLUDES SUBROUTINES SEQ1,TRACD,GETTR,RDBUF,RREQK,
20:     C RRTRC,RQAHM AND RQITL, WAS WRITTEN BY
21:     C PETER R. STEVENSON OF THE U.S.G.S.
22:     C IN JANUARY 1979
23:     C - - - - -
24:     C
25:     C
26:     C TBUF = TEMPORARY STORAGE AREA
27:     C EVLST = EVENT DIRECTORY INFORMATION
28:     C TRINDX = TRACE INDEX
29:     C DRFILE = LOGICAL UNIT OF DIRECTORY FILE
30:     C
31:     PARAMETER LSZ = 43
32:     COMMON/EVCOM/TBUF(256),EVLST(LSZ),ICURR
33:     INTEGER TBUF,EVLST,ENAM(40),TIME(7),EVENT(5),TRACE
34:     *,UNQID(4),TRINDX(5),DRFIL
35:     EQUIVALENCE (EVLST(41),TRINDX(1)),(EVLST(22),TIME(1)),
36:     *(EVLST(2),ENAM(1)),(EVLST(29),DELT),(EVLST(32),DUR)
37:     C
38:     PARAMETER DRFIL = 14
39:     C
40:     INCLUDE 'C$IOCOM'
41:     C
42:     COMMON /IOCOM/MKNT,IFIRST,LSIZ,LL,NM,IFLAG,JFLAG,LBLK,RCNT,FCNT
43:     PARAMETER MBLK = 4, NSIZ = MBLK*256, IFILE = 12
44:     C
45:     C
46:     STATIC JKNT
47:     DIMENSION LBUF(1)
48:     COMMON /IOBUF/HBUF(NSIZ)
49:     C
50:     C IFIRST = FIRST WORD TO RETRIEVE
51:     C LKNT = USER WORD COUNTER
52:     C MKNT = INPUT WORD COUNTER
53:     C LL = USER BUFFER COUNTER
54:     C NM = INPUT BUFFER COUNTER
55:     C LSIZ = LENGTH OF USER BUFFER
56:     C NSIZ = LENGTH OF INPUT BUFFER
57:     C IFLAG = 0 FOR EMPTY BUFFER, 1 FOR FILLED BUFFER
58:     C JFLAG = 0 FOR NORMAL, 1
59:     C JSIZ = EFFECTIVE LENGTH OF INPUT BUFFER

```

```

60: C
61: C
62: C
63:     LKNT = 0
64:     LL = LL + 1
65: C   USER BUFFER LOOP
66: C   EXIT WHEN USER BUFFER FULL
67: C
68: 1   CONTINUE
69: C
70: C   IS INPUT BUFFER FULL?
71: C   IF IT IS, THEN RESET COUNT TO ZERO
72: C
73:     IF(NKNT.EQ.NSIZ)IFLAG = 0
74:     IF(NKNT.EQ.NSIZ)NKNT = 0
75: C
76: C   IS INPUT BUFFER FLAG ZERO?
77: C
78:     IF(IFLAG.NE.0)GOTO 2
79: C
80: C   IF SO THEN BUMP BUFFER COUNTER
81: C   AND READ ANOTHER BUFFER
82: C
83:     NM = NM + 1
84:     CALL RDBUF(NBUF,NM,NBLK,IFILE,IERR)
85: C   CHECK FOR EOF
86:     IF(IERR.EQ.11)GOTO 500
87: C   RESET BUFFER FLAG TO FULL
88:     IFLAG = 1
89: C   CHECK FOR LAST BLOCK
90:     IF(IERR.GT.0)GOTO 2
91:     JJ = IABS(IERR)
92:     JFLAG = 1
93:     JKNT = JJ*256
94: C
95: C   BUMP INPUT/OUTPUT WORD COUNTS
96: C
97: 2   LKNT = LKNT + 1
98:     NKNT = NKNT + 1
99: C   BUMP TOTAL WORD COUNT (9-7-79)
100:     RCNT = RCNT + 1
101: C   COMPUTE INDEX OF WORD RELATIVE TO INPUT BUFFER
102: C   MOVE WORD( USER INDEX IS THE SAME AS THE USER WORD COUNT )
103:     LBUF(LKNT) = NBUF(NKNT)
104: C   CHECK FOR LAST VALID POINT OF LAST BLOCK
105:     IF(JFLAG.EQ.1.AND.JKNT.EQ.NKNT)GOTO 500
106: C   DON' GO BEYOND LAST VALID POINT AS COMPUTED FROM
107: C   DUR,DELT AND ISTART (9-7-79)
108:     IF(RCNT.GE.FCNT)GOTO 500
109: C
110: C   CHECK FOR FULL USER(OUTPUT) BUFFER
111: C   IF NOT FULL THEN GO GET ANOTHER WORD
112:     IF(LKNT.NE.LSIZ)GOTO 1
113:     IEND=0
114:     RETURN
115: C
116: C   EOF/END-OF-BLOCK RETURN
117: 500  IEND = 1
118:     RETURN
119:     END

```

ECLIPSE FORTRAN 5, VERSION 5.21 -- MONDAY, AUGUST 11, 1980 12:16:37 PM

RDBUF.FR

```

1)      SUBROUTINE RDBUF(NBUF,NN,NBLK,IFILE,IERR)
2)      C
3)      C   THIS ROUTINE INTERFACES GETTR TO RDBLK (D.G.C. RUNTIME ROUTINE)
4)      C
5)      C
6)      COMMON /TRCOM/NAM(2),IFAZ,NSTRT,NEND,NTR
7)      INTEGER FBLK
8)      C
9)      DIMENSION NBUF(1)
10)     C
11)     C   NN      = NUMBER OF INPUT BUFFERS READ
12)     C   NBLK   = NUMBER OF DISK BLOCKS PER BUFFER
13)     C   NSTRT  = FIRST DISK BLOCK FOR CURRENT TRACE
14)     C
15)     FBLK = (NN-1)*NBLK + NSTRT
16)     NBLK = NBLK
17)     IF(NEND.LE.(FBLK+NBLK-1))NBLK = NEND - FBLK + 1
18)     C   INPUT DISK BLOCK
19)     CALL RDBLK(IFILE,FBLK,NBUF,NBLK,IERR)
20)     IF(IERR.EQ.1.OR.IERR.EQ.11)GOTO 1
21)     C   ERROR PATH COMES HERE
22)     TYPE 'READ ERROR @ ',IERR,' FBLK=',FBLK
23)     RETURN
24)     C   NORMAL PATH
25)     1   CONTINUE
26)     C   CHECK FOR END OF TRACE
27)     IF(FBLK+NBLK-1.EQ.NEND)IERR = -NBLK
28)     RETURN
29)     END

```

ECLIPSE FORTRAN 5, VERSION 5 21 -- MONDAY, AUGUST 11, 1980 12:17:16 PM

RQEOK.FR

```

1:      SUBROUTINE RQEOK(LIST,EVENT)
2:      C
3:      DIMENSION LIST(1)
4:      INCLUDE 'C$EVCOM'
5:      C -----
6:      C THE SEQUENTIAL ACCESS PACKAGE, WHICH
7:      C INCLUDES SUBROUTINES SEQ1,TRACD,GETTR,RDBUF,RQEOK,
8:      C RQTRC,RQNAM AND RQTTL, WAS WRITTEN BY
9:      C PETER R. STEVENSON OF THE U.S.G.S.
10:     C IN JANUARY 1979
11:     C -----
12:     C
13:     C
14:     C TBUF = TEMPORARY STORAGE AREA
15:     C EVLST = EVENT DIRECTORY INFORMATION
16:     C TRINDX = TRACE INDEX
17:     C DRFILE = LOGICAL UNIT OF DIRECTORY FILE
18:     C
19:     C PARAMETER LSZ = 45
20:     C COMMON, EVCOM/TBUF(256),EVLST(LSZ),ICURR
21:     C INTEGER TBUF,EVLST,ENAM(40),TIME(7),EVENT(5),TRACE
22:     C * ,UNQID(4),TRINDX(5),DRFIL
23:     C EQUIVALENCE (EVLST(41),TRINDX(1)),(EVLST(22),TIME(1)),
24:     C *(EVLST(2),ENAM(1)),(EVLST(29),DELT),(EVLST(32),DUR)
25:     C
26:     C PARAMETER DRFIL = 14
27:     C
28:     C OPEN FILES
29:     C CALL SEQ1(EVENT)
30:     C READ EVENT HEADER BLOCK
31:     C NREC = 1
32:     C READ(DRFIL,REC=NREC)TBUF
33:     C ICURR = NREC
34:     C
35:     C LOAD EVENT DIRECTORY ARRAY
36:     C DO 1 J=1,LSZ
37:     C   EVLST(J) = TBUF(J)
38:     C
39:     C ASSIGN ENTRIES IN EVLST FROM EVENT RECORD
40:     C
41:     C LIST(1) = EVLST(1)
42:     C
43:     C TIME, INCLUDING SECOND(REAL)
44:     C
45:     C DO 3 J=1,7
46:     C   LIST(J+1) = TIME(J)
47:     C
48:     C DURATION(REAL)
49:     C
50:     C LIST(9) = EVLST(32)
51:     C LIST(10) = EVLST(33)
52:     C
53:     C DELTA-T (REAL)
54:     C
55:     C LIST(11) = EVLST(29)
56:     C LIST(12) = EVLST(30)

```



```

57: C
58: C   EVENT ID (UNPACKED ASCII CHARACTERS)
59: C
60: C   DO 5 J=1,5
61: S   LIST(12+J) = EVLST(33+J)
62: C
63: C   NUMBER OF TRACES FOR THIS EVENT
64: C
65: C   LIST(18) = EVLST(31)
66: C
67: C   RETURN
68: C   END

```

ECLIPSE FORTRAN 5, VERSION 5.21 -- MONDAY, AUGUST 11, 1980 12:19:27 PM

RQTTL.FR

```

1:   SUBROUTINE RQTTL(EVTNAM)
2:   INTEGER EVTNAM(20)
3: C   THIS ROUTINE GETS EVENT TITLE FOR THIS EVENT
4: C
5:   INCLUDE 'C$EVCON'
6: C
7: C   -----
8: C   THE SEQUENTIAL ACCESS PACKAGE, WHICH
9: C   INCLUDES SUBROUTINES SEQ1,TRACO,GETTR,RDBUF,RREQK,
10: C   RQTRC,RQNAM AND RQTTL, WAS WRITTEN BY
11: C   PETER R. STEVENSON OF THE U.S.G.S.
12: C   IN JANUARY 1979
13: C   -----
14: C
15: C   TBUF = TEMPORARY STORAGE AREA
16: C   EVLST = EVENT DIRECTORY INFORMATION
17: C   TRINDX = TRACE INDEX
18: C   DRFILE = LOGICAL UNIT OF DIRECTORY FILE
19: C
20: C   PARAMETER LSZ = 45
21: C   COMMON/EVCON/TBUF(256),EVLST(LSZ),ICURR
22: C   INTEGER TBUF,EVLST,ENAM(40),TIME(7),EVENT(5),TRACE
23: C   *,UNQID(4),TRINDX(5),DRFIL
24: C   EQUIVALENCE (EVLST(41),TRINDX(1)),(EVLST(22),TIME(1)),
25: C   *(EVLST(2),ENAM(1)),(EVLST(29),DELT),(EVLST(32),DUR)
26: C
27: C   PARAMETER DRFIL = 14
28: C
29: C   LOAD UP NAME ARRAY
30: C   DO 1 J=1,20
31: I   EVTNAM(J) = ENAM(J)
32: C   RETURN
33: C   END

```

ECLIPSE FORTRAN 5, VERSION 5 21 -- TUESDAY, FEBRUARY 17, 1981 9:46:53 AM

RQTRC.FR

```

1:      SUBROUTINE RQTRC(INAM,TRACE)
2:      COMMON /TRCOM/TRBLK(5),NTR
3:      C
4:      C - - - - MODIFIED 2-17 81 TO FIX BUG
5:      C 1ST TRACE WAS RETRIEVED WHEN 51ST REQUESTED
6:      C - - - - MODIFIED 9/14/79 BY P.R. STEVENSON TO FIX BUG
7:      C      JOFF WAS BEING CALCULATED USING TRACE PATHER
8:      C      THAN JTRACE
9:      C
10:     C   INAM = STATION NAME (4 CHAR. ASCII)
11:     C   EVENT = EVENT NUMBER
12:     C   TRACE = REQUESTED TRACE NUMBER
13:     C
14:     C       INTEGER TRBLK
15:     C
16:     C       INCLUDE 'C$EVCOM'
17:     C - - - - -
18:     C THE SEQUENTIAL ACCESS PACKAGE, WHICH
19:     C INCLUDES SUBROUTINES SEQ1,TRACD,GETTR,RDBUF,RQEQK,
20:     C RQTRC,RQNAM AND RQTL, WAS WRITTEN BY
21:     C PETER R. STEVENSON OF THE U.S.G.S.
22:     C IN JANUARY 1979
23:     C - - - - -
24:     C
25:     C
26:     C   TBUF = TEMPORARY STORAGE AREA
27:     C   EVLST = EVENT DIRECTORY INFORMATION
28:     C   TRINDX = TRACE INDEX
29:     C   DRFILE = LOGICAL UNIT OF DIRECTORY FILE
30:     C
31:     C   PARAMETER LSZ = 45
32:     C   COMMON/EVCOM/TBUF(256),EVLST(LSZ),ICURR
33:     C   INTEGER TBUF,EVLST,ENAM(40),TIME(7),EVENT(5),TRACE
34:     C   *,UNQID(4),TRINDX(5),DRFIL
35:     C   EQUIVALENCE (EVLST(41),TRINDX(1)),(EVLST(22),TIME(1)),
36:     C   *(EVLST(2),ENAM(1)),(EVLST(29),DELT),(EVLST(32),BUR)
37:     C
38:     C   PARAMETER DRFIL = 14
39:     C
40:     C   DIMENSION INAM(2)
41:     C
42:     C   COMPUTE LOCATION OF TRACE DIRECTORY RECORD
43:     C   IBPT = (TRACE-1)/51 + 1
44:     C
45:     C   IS BLOCK ALREADY IN COMPUTER MEMORY
46:     C
47:     C   NREC = TRINDX(IBPT)
48:     C
49:     C   JTRACE = MOD(TRACE,51)
50:     C 51 REPLACED 1 - FIX OF 2-17-81
51:     C   IF(JTRACE.EQ.0)JTRACE = 51
52:     C
53:     C
54:     C   IF(NREC.EQ.ICURR)GOTO 1
55:     C
56:     C   IF NOT CURRENT BLOCK THEN READ BLOCK

```

```

57: C
58: READ(DRFIL,REC=MREC)TBUF
59: C
60: C COMPUTE OFFSET TO TRACE DIRECTORY
61: 1 JOFF = (JTRACE-1)*5
62: C LOAD THE TRACE DIRECTORY INTO COMMON
63: DO 2 J=1,5
64: 2 TRBLK(J)=TBUF(JOFF+J)
65: NTR=TRACE
66: INAM(1) = TRBLK(1)
67: INAM(2) = TRBLK(2)
68: C
69: RETURN
70: END

```

ECLIPSE FORTRAN 5, VERSION 5.21 -- TUESDAY, MARCH 17, 1981 10:31:18 AM

RQTN.FR

```

11 SUBROUTINE RQTN(TRNAM, ITR, NTR, IER)
12 C PETER R. STEVENSON 3-26-80
13 INTEGER TRNAM(2), INAM(2)
14 IER = 0
15 DO 1 J=1, NTR
16 CALL RQTRC(INAM, J)
17 IF(INAM(1).NE.TRNAM(1))GOTO 1
18 IF(INAM(2).NE.TRNAM(2))GOTO 1
19 ITR = J
20 GOTO 2
21 1 CONTINUE
22 IER = 1
23 TYPE 'STATION NOT FOUND'
24 2 RETURN
25 END
26

```

ECLIPSE FORTRAN 5, VERSION 5.21 -- MONDAY, AUGUST 11, 1980 12:18:49 PM

RQAN.FR

```

11 SUBROUTINE RQAN(ISTA, MREQ, IREQ)
12 DIMENSION ISTA(2,32), IREQ(1), INAM(2)
13 C THIS ROUTINE GETS ALL STATION NAMES FOR THIS EVENT
14 C
15 DO 1 J=1, MREQ
16 KK = IREQ(J)
17 CALL RQTRC(INAM, KK)
18 ISTA(1, J) = INAM(1)
19 1 ISTA(2, J) = INAM(2)
20 RETURN
21 END

```

ECLIPSE FORTRAN 5, VERSION 5.21 -- THURSDAY, MARCH 26, 1981 9:32:35 AM

ROFAZ.FR

```

1:      SUBROUTINE ROFAZ(TRNAM,IFZ,ICARD,SCARD,ISUMM,IERR)
2:      C      PETER R. STEVENSON 8-13-80
3:      C      MODIFIED 2-12-81 TO RETURN STATION CARD P.R.S.
4:      INTEGER TRNAM(2),EOS,EOF,NAM(2),ICARD(40),ISUMM(40),SCARD(40)
5:      C      GET CONTROL POINTERS
6:      READ(1,100,REC=1,END=550)EOF,EOS
7:      100  FORMAT(35X,214)
8:      C      READ SUMMARY CARD
9:      READ(1,101,REC=2,END=500)ISUMM
10:     C      SEARCH STATION LIST FOR THE STATION
11:     JEOS = EOS - 1
12:     DO 5 J=4,JEOS
13:     READ(1,101,REC=J,END=602,ERR=500)SCARD
14:     DECODE(SCARD,103)NAM
15:     103  FORMAT(2A2)
16:     GOTO 6
17:     602  TYPE 'EOF ON READING STATION RECORD, J=',J
18:     GOTO 500
19:     6    CONTINUE
20:     IF(NAM(1) .NE. TRNAM(1))GOTO 5
21:     IF(NAM(2) .NE. TRNAM(2))GOTO 5
22:     GOTO 7
23:     5    CONTINUE
24:     7    CONTINUE
25:     C      -----
26:     C      SEARCH PHASE LIST FOR THE STATION
27:     IEOS=EOS+1
28:     IEOF=EOF-1
29:     DO 1 J=IEOS,IEOF
30:     READ(1,101,REC=J,END=600,ERR=499)ICARD
31:     101  FORMAT(40A2)
32:     DECODE(ICARD,102)NAM,JFZ
33:     102  FORMAT(2A2,2X,A2)
34:     GOTO 601
35:     600  TYPE 'END OF FILE ON .EQ, J=',J
36:     GOTO 500
37:     601  CONTINUE
38:     IF(NAM(1) .NE. TRNAM(1))GOTO 1
39:     IF(NAM(2) .NE. TRNAM(2))GOTO 1
40:     IF(JFZ .NE. IFZ)GOTO 1
41:     GOTO 2
42:     1    CONTINUE
43:     GOTO 500
44:     499  TYPE 'ERROR READING PHASE CARD. REC=',J
45:     500  IERR = -1
46:     RETURN
47:     2    IERR = 0
48:     RETURN
49:     550  TYPE 'NO 1ST RECORD IN .EQ FILE'
50:     IERR = -1
51:     RETURN
52:     END
53:

```

ECLIPSE FORTRAN 5, VERSION 5.21 -- FRIDAY, MARCH 27, 1981 4:39:53 PM

VIEW.FR

```

1: C PROGRAM TO VIEW SPRINT FILES
2: C ON THE TEKTRONIX 4014
3: C
4: C PETER R. STEVENSON
5: C 3/27/81
6: C
7: C MEETS 'NEU' SPRINT FORMAT SPECS.
8: C
9: C INTEGER TS(3100),ID(5),TX,TY, TXL,TYL,LLL(21)
10: C REAL T(3)
11: C DATA TX,TY,TXL,TYL/404,303,3498,2020/
12: C YN = -512
13: C YR = 1024
14: C IC = 2
15: C TIC = 10.
16: C
17: C OPEN 4,'STT01'
18: C OPEN 5,'STT11'
19: C CALL INITT(0)
20: C CALL TERM(3,4096)
21: C CALL SETBUF(3)
22: C CALL CHRSIZE(4)
23: C CALL TSEND
24: C CALL NITEK(1,0)
25: C LU = 0
26: C OPEN LU,'SPRINT'
27: C
28: C 1 CONTINUE
29: C N1 = 3100
30: C CALL FSEIS(TS,N1, ID,SNAM, T,DT,DD,OR,NAG, IFN,JFN,EPD,AZ,AIM
31: C ,PT,ST,LU,0)
32: C CALL ERASE
33: C XM = 0.
34: C XR =N1*DT
35: C SETUP TIME SERIES WINDOW
36: C CALL VPORT(XM,XR,YN,YR,TX,TXL,TY,TYL,TIC,IC)
37: C LABELS
38: C LX = 404
39: C LY = 2350
40: C CALL NOVASS(LX,LY)
41: C ENCODE(LLL,100)SNAM
42: C 100 FORMAT(A4)
43: C CALL ADUTST(4,LLL)
44: C CALL NOVREL(100,0)
45: C ENCODE(LLL,102)ID
46: C 102 FORMAT('EVENT: ',4A2,A1)
47: C CALL ADUTST(16,LLL)
48: C
49: C PLOT TIME SERIES
50: C CALL ITRACE(TS,N1 XM,DT)
51: C
52: C CALL NOVASS(10,10)
53: C CALL TSEND
54: C READ(11,400)LLL(1)
55: C 400 FORMAT(S1)
56: C LSS = 51400K

```

```
57:      IF(LLL(1).NE.L88)GOTO 1
58:      STOP
59:      END
```

ECLIPSE FORTRAN 5, VERSION 5.21 -- THURSDAY, MARCH 12, 1981 9:48:20 AM

ITRACE.FR

```
1:      SUBROUTINE ITRACE(A,NL,S,DX)
2:      INTEGER A(NL)
3:      C ROUTINE TO PLOT ONE TRACE USING TEKTRONIX PACKAGE
4:      C P.R.STEVENSOM 11-25-80
5:      C A = AMPLITUDE
6:      C NL = NUMBER OF POINTS
7:      C S = STARTING X COORDINATE
8:      C DX = VALUE OF EACH X INCREMENT
9:      C
10:     R = A(1)
11:     CALL MOVEA(S,R)
12:     DO 1 J=2,NL
13:     X = J*DX
14:     R = A(J)
15:     CALL DRAWA(X,R)
16: 1    CONTINUE
17:     CALL TSEND
18:     RETURN
19:     END
```

ECLIPSE FORTRAN 9, VERSION 5.21 -- FRIDAY, MARCH 27, 1991 4:22:27 PM

FSEIS.FR

```

1:      SUBROUTINE FSEIS(X,NPTS,ID,SNAM,T,DT,DB,OR,MAG,IFM,JFN,
2:      *EPD,AZ,AIN,PT,ST,LU,ISW)
3:      C SUBROUTINE TO READ TEXT FILES GENERATED
4:      C BY ( . . . ) OPTION OF SISDS
5:      C
6:      C P.R. STEVENSON 11-29-89
7:      C MODIFIED 2-13-91 TO ACCOMODATE NEW SPRINT FORMAT P.R.S.
8:      C
9:      C X = DATA ARRAY
10:     C NPTS = NUMBER OF POINTS TO BE OUTPUT
11:     C (ON INPUT IT IS THE MAXIMUM
12:     C PERMISSIBLE NUMBER TO STORE IN X)
13:     C ID = EVENT IDENTIFICATION
14:     C SNAM = STATION NAME
15:     C T = TIME OF START OF DATA(HR,MIN,SEC)
16:     C DT = SAMPLING INTERVAL
17:     C DB = STATION ATTENUATION SETTING(DECIBELS)
18:     C OR = ORIGIN TIME(HR,MIN,SEC) AND PLACE(LAT,LAT-MIN,LONG,LONG MIN)
19:     C MAG = ESTIMATED MAGNITUDE
20:     C IFM = P-WAVE FIRST MOTION DIRECTION
21:     C JFN = S-WAVE FIRST MOTION DIRECTION
22:     C EPD = EPICENTRAL DISTANCE
23:     C AZ = AZIMUTH
24:     C AIN = ANGLE OF INCIDENCE
25:     C PT = P-ARRIVAL TIME (HR,MIN,SEC)
26:     C ST = S-ARRIVAL TIME (HR,MIN,SEC)
27:     C LU = LOGICAL UNIT OF INPUT FILE
28:     C ISW = MESSAGE SWITCH, ISW=0 FOR NO MESSAGE, ISW=1
29:     C FOR PRINT OUT INPUT DATA TO MESSAGE FILE.
30:     INTEGER X
31:     DIMENSION X(1),ID(5),T(3),OR(7),ICARD(40),PT(3),ST(3)
32:     INTEGER DUM,DB,AIN,AZ
33:     REAL MAG
34:     NMAX = NPTS
35:     NREN = 0
36:     C
37:     ISKIP = 2H-
38:     IP = 2HP
39:     IS = 2HS
40:     ILIN = 2H--
41:     READ(LU,100,END=500)ID,T,DT
42:     100 FORMAT(5X,5A2,10X,2F2.0,2X,F0.3,2X,F5.3)
43:     READ(LU,101,END=500)SNAM,NPTS
44:     101 FORMAT(10X,A4,14X,I0)
45:     C
46:     C MODS OF 2-13-91 TO ACCOMODATE NEW SPRINT FORMAT
47:     READ(LU,103,END=500)OR,MAG
48:     103 FORMAT(7X,2F2.0,F6.2,F3.0,1X,F5.2,F4.0,1X,F5.2,8X,F6.2)
49:     READ(LU,104,END=500)DB
50:     104 FORMAT(31X,I3)
51:     DO 5 J=1,2
52:     READ(LU,105)ICARD
53:     105 FORMAT(40A2)
54:     IF(ICARD(1).EQ.ISKIP)GOTO 10
55:     IF(ICARD(4).EQ.IP)DECODE(ICARD,106)IFM,PT,EPD,AZ,AIN
56:     IF(ICARD(4).EQ.IS)DECODE(ICARD,106)JFN,ST,EPD,AZ,AIN

```

```

57: 106  FORMAT(0X,A1,10X,2F2.0,1X,F5.2,17X,F6.2,1X,13,1X,13)
58: 5      CONTINUE
59:      READ(LU,105)ICARD
60: 10    READ(LU,105)ICARD
61:      IF(ICARD(1).NE.ILIN)WRITE(10,105)ICARD
62:  C
63:  C - END OF HOOS -
64:  C
65:      IF(NPTS.GT.NMAX)NREN = NPTS - NMAX
66:      IF(NPTS.GT.NMAX)TYPE '--- ',NPTS,' POINTS READ. ---'
67:      IF(NPTS.GT.NMAX)TYPE '*** TOO MANY POINTS.',NREN,' POINTS SKIPPED! ***'
68:      IF(NPTS.GT.NMAX)NPTS = NMAX
69:      READ(LU,102,END=500)(X(J),J=1,NPTS)
70: 102   FORMAT(5X,15I5)
71:  C
72:      IF(ISW.EQ.0)GOTO 90
73:      WRITE(10,1000)ID,SNAM
74: 1000  FORMAT('ID=',5A2,' SNAM=',A4)
75:      TYPE 'NPTS=',NPTS
76:      WRITE(10,1001)(X(J),J=1,NPTS)
77: 1001  FORMAT('X=',/,5X,15I5)
78: 90    IF(NREN.NE.0)GOTO 92
79:  C READ END OF TRACE DELIMITER
80:      READ(LU,105)ICARD
81:      RETURN
82: 92    CONTINUE
83:  C DUMMY READ OF EXCESS DATA
84:      IEND = 2NEN
85: 91    CONTINUE
86:      READ(LU,105,END=500)ICARD(1)
87:      IF(IEND.NE.ICARD(1))GOTO 91
88:      RETURN
89: 500   TYPE '0/0/0 END OF DATA FILE 0/0/0'
90:      STOP
91:      END

```


ECLIPSE FORTRAN 9, VERSION 5.21 -- MONDAY, NOVEMBER 24, 1988 10:06:07 AM

VPORT.FR

```

11      SUBROUTINE VPORT(R1,RL,S1,SL,IX,LX,IY,LY,TIC,IC)
12      DIMENSION SXB(2),SXL(2),SBB(2)
13      CC
14      CC IF IC=0, DRAW BOX ONLY
15      CC IF IC=1, PUT ON TIC MARKS AT TIC INTERVALS
16      CC STARTING AT FIRST UNIT INTERVAL
17      CC IF IC=2, ALSO PUT ON FINE TICS AT TIC/10
18      CC INTERVALS
19      CC
20      S2=S1+SL
21      R2=R1+RL
22      S=S2/LY
23      CC
24      CALL VWINDO(R1,RL,S1,SL)
25      CALL BOX(IX,LX,IY,LY)
26      IF(IC.EQ.0)GO TO 99
27      TT=TIC/10
28      Y=R1/TIC+1
29      Z=INT(Y)*TIC
30      NTIC=RL/TIC+1
31      CC
32      SXB(1)=-60*S
33      SXB(2)=60*S
34      SXL(1)=-40*S
35      SXL(2)=40*S
36      SBB(1)=S2
37      SBB(2)=S1
38      CC
39      DO 20 L=1,2
40      CC
41      SB=SBB(L)
42      SIT=SXB(L)
43      SLT=SXL(L)
44      IF(IC.EQ.1)GO TO 6
45      DO 5 K=1,9
46      XTT=2-K*TT
47      IF(XTT.LE.R1)GO TO 6
48      SLT=SXL(L)
49      CALL MOVEA(XTT,SB)
50      CALL DRAWR(0.,SLT)
51      S
52      CC
53      6      CONTINUE
54      CC
55      DO 1 J=1,NTIC
56      XT=Z+(J-1)*TIC
57      CALL MOVEA(XT,SB)
58      CC LARGE TIC DRAWN DOUBLE
59      CALL DRAWR(0.,SIT)
60      CALL DRAWR(0.,-SIT)
61      IF(IC.EQ.1)GO TO 1
62      CC
63      1      DRAW SMALL TICS
64      DO 2 K=1,9
65      XTT=XT+TT*K
66      IF(XTT.GE.R2)GO TO 1

```

```
57:      CALL MOVEA(XTT,SB)
58:      CALL DRAW(O.,SLT)
59:      2      CONTINUE
60:      CC
61:      1      CONTINUE
62:      20     CONTINUE
63:      CC
64:      99     RETURN
65:      END
```