Best
Available
Copy

LEVEL II

ADA058532

AD No. ____
DDC FILE COPY



Computer Corporation of America



575 Technology Square
Cambridge
Massachusetts 02139
617-491-3670

# Program Design for SWF-D: The Signal Waveform File Demon.

Donald E. Eastlake, III
Joanne Z. Sattley

Technical Report, CCA-78-10

June 30, 1978







This research was supported by the Advanced Research Projects Agency of the Department of Defense under Contract No. N00039-78-C-0246, ARPA Order No. 3175-6. The views and conclusions contained in this document 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 or the U.S. Government.

## Table of Contents

# 1. Overview

The seismic data activity sponsored by the Nuclear Monitoring Research Office (NMRO) involves the collection, storage and processing of seismic waveform information as measured by instruments installed throughout the world. The data activity is intended to assist seismologists in exploring techniques for the detection of seismic events, for pinpointing their location, and for recognizing the causes of these events.

The Datacomputer is the primary storage and retrieval resource being utilized by the effort. The Datacomputer provides the facility required to store the very large amounts of on-line data, and the database management tools to allow users of seismic data to retrieve manageable portions of a database for analysis.

At the present time, approximately 85 billion bits of raw seismic data are stored on-line with another 164 billion bits stored off-line; seismic data from six to twelve weeks old is directly accessible on-line. Of these, the seismic data files that are expected to be the most useful for seismological research are those which contain summaries of significant events, and the corresponding signal waveforms.

The amount of event summary information available is steadily accumulating, thus increasing the range of experiments possible with the data. The event summary files (ESF) are sufficiently small that they will all be retained on-line indefinitely.

To increase the span of the on-line basic data window from the current nearly two-month period to an anticipated twelve-month period, a file of only the seismic readings that are directly associated with events in the event summary list was planned. The basic foundation for this set of files, the Signal Waveform Files (SWF), was laid out in August 1977, jointly by representatives of ARPA-NMRO, Vela Seismological Center (VSC), Lincoln Laboratories Applied Seismology Group (LL-ASG), and the Seismic Data Analysis Center (SDAC).

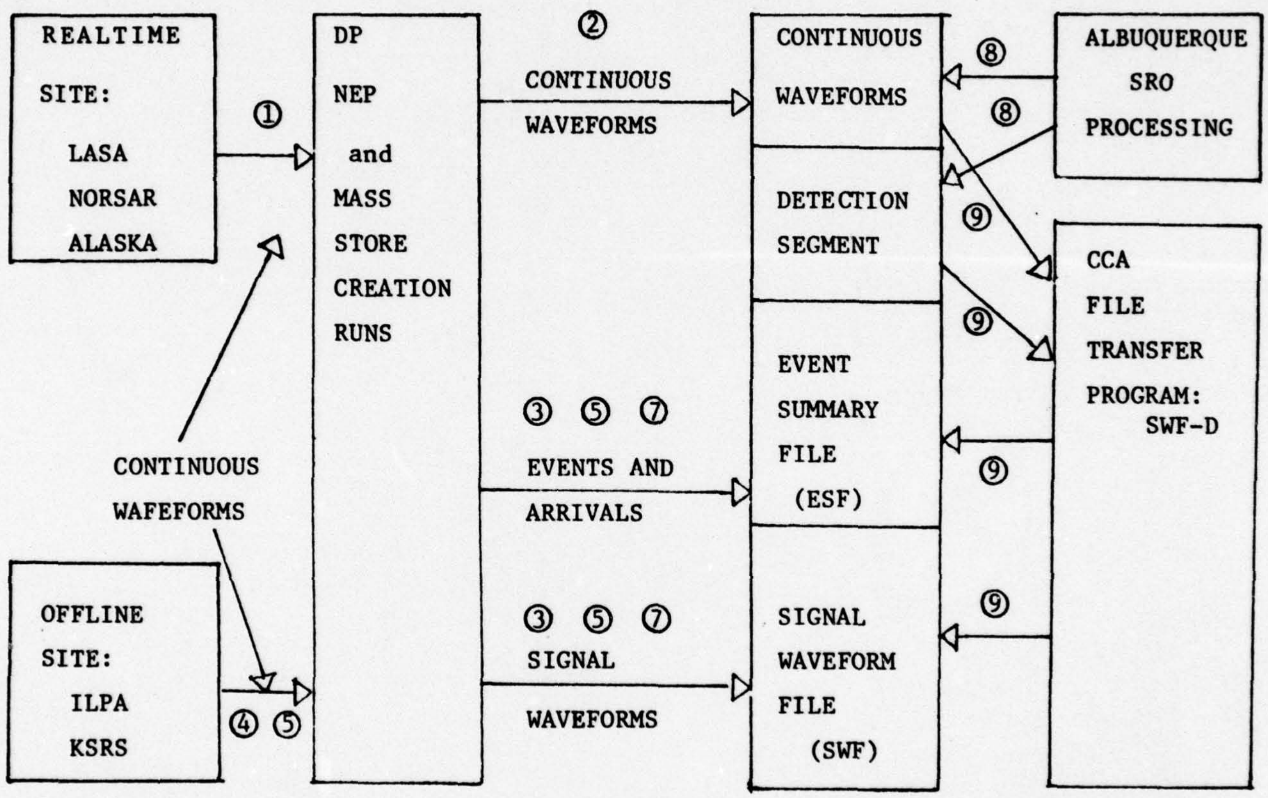
Seismic data to be stored in the Signal Waveform Files falls into two general categories. First, there are data readings that do not exist in any other form on the Datacomputer. These data will be stored directly into the waveform file by SDAC in segments corresponding to entries in the event files.

Second, some of the data segments that belong in the Signal Waveform Files already exist on the Datacomputer, embedded in the basic seismic data files. It is appropriate for these data to simply be copied into the waveform files when seismic events are detected and noted in the Event Summary File. (The indications in the Event Summary File are computed by SDAC). These data segments will be appended to the waveform file by the program described in this document.

The following chart, figure 1.1, depicts the overall data flow through the system, with an indication of the order of processing. It is based upon a diagram furnished by Teledyne at the 13 April 1978 SWF Design Review Meeting held at SDAC.

Data Flow

Figure 1.1




1 - Waveform data from realtime sites arrives at SDAC

2 - Some data from realtime site goes to Datacomputer

3 - ESF/SWF including realtime site signal waveforms go to Datacomputer

4 - ILPA field tape arrives at SDAC

5 - ILPA signal waveforms go to Datacomputer

6 - KSRS field tape arrives at SDAC

7 - KSRS signal waveforms go to Datacomputer

8 - Albuquerque reviewed SRO data goes to Datacomputer

9 - CCA FILE TRANSFER PROGRAM completes signal waveforms

## 2. Introduction

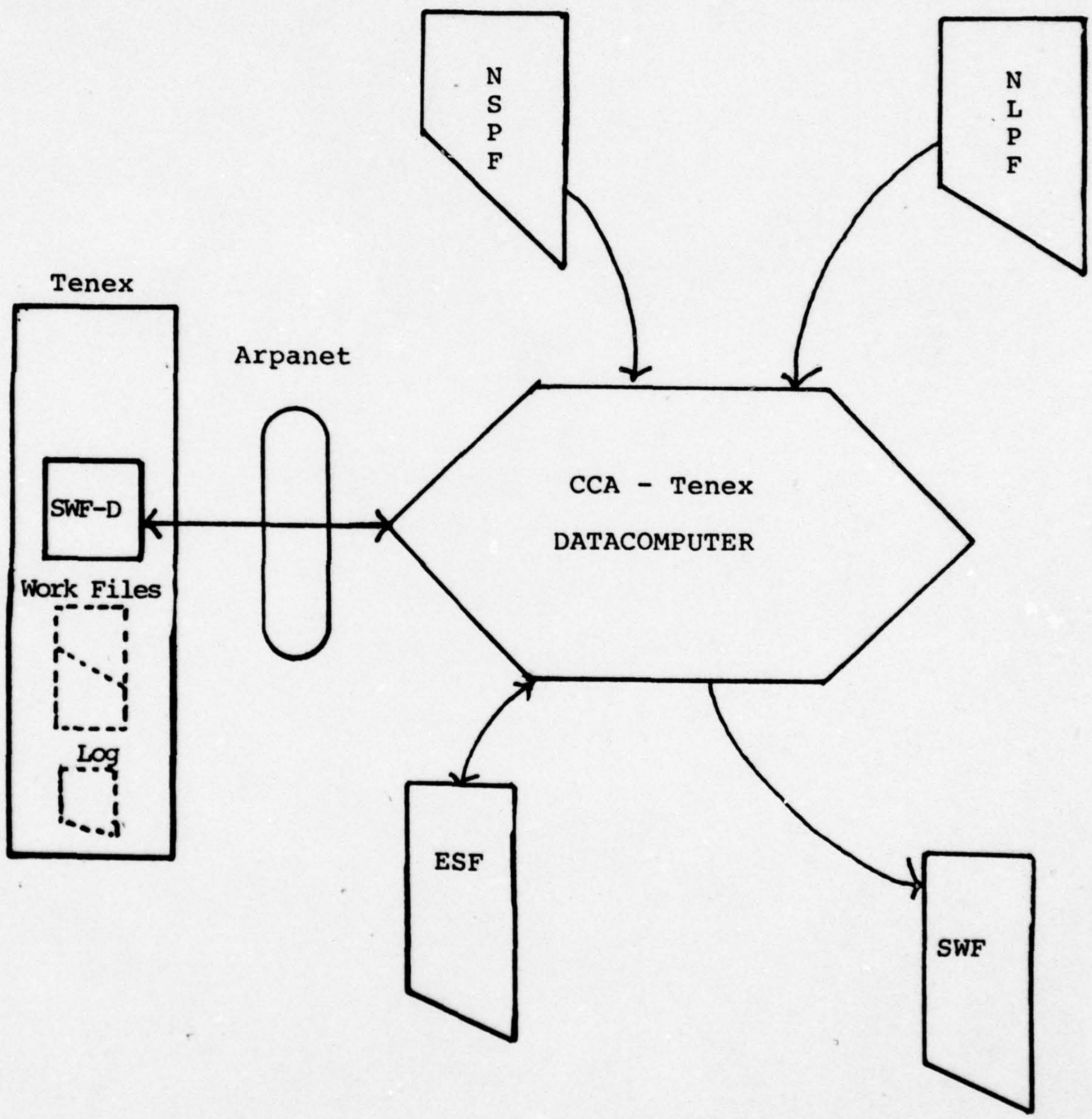
This document describes plans for the creation of a program which will assist in the generation of the Signal Waveform File. The program will operate under the direction of flags set in a seismic event summary file by the Seismic Data Analysis Center (SDAC) in Alexandria, Virginia [TELEDYNE].

Figure 2.1 depicts the interrelationship of the CCA program and the Datacomputer to the seismic files which are involved.

Section 4 presents a detailed description of the data which will be referenced. Section 3 below discusses the tasks that the CCA program is expected to perform, and some techniques developed for accomplishing them. Section 5 describes the program's components. Appendix A contains a Datalanguage definition of the new Datacomputer ports which will be created.

CCA & DC relationship with seismic files

Figure 2.1



## 3. Concepts

The CCA program is designed to be reliable, efficient, and responsive to its working environment. It will operate on CCA-Tenex in background mode without scheduled operator action or intervention.

The SWF-D ("D" for "Demon") program will be written in the BCPL Compiler Language, a high-level language maintained for Tenex by BBN. Some of CCA's recent software work [CCAd] has been done using this language, and its use as a programming tool has met with general approval.

The program will communicate with the Datacomputer via the Arpanet in the manner of a normal Datacomputer user. We will make use of an existing package of subroutines to accomplish the network transmissions[CCAa]. This interface package was originally developed at CCA and is currently being maintained by CCA personnel.

Because SWF-D will operate on the same computer system as the DC-203 Datacomputer, it will need to avoid imposing a crushing load on either Tenex or the Datacomputer.

Facilities for inspecting Tenex system status exist, and will be incorporated into the startup control logic, so that during high-load situations, SWF-D will not start up.

In order to retrieve Datacomputer status information, SWF-D will make use of an existing status checker program [CCAb]. The information obtained will be used to inhibit operation if the Datacomputer is heavily loaded, or if TBM access is suspended, or if the SIP is running.

The program will reliably resume operation after operating system failure or a period of inaccessibility of the DC-203 Datacomputer at CCA, and will make use of the task status information last recorded on its operation log to resume running from where it left off. In the case of a Tenex crash, the program will automatically be restarted by Tenex as part of its initialization sequence.

If the Datacomputer is unavailable, or if the net connection fails during regular program execution, SWF-D will suspend its operations for a computed interval of time. It has been anticipated that, after the program has been placed in service, some re-tuning of this computation will be necessary -- and, to facilitate this, we've provided for experimentally adjusting the periodicity.

In order to recognize the non-availability or partial availability of the seismic data desired, we are providing two kinds of tests, one which applies to long-period SRO data, the other, to short-period SRO data.

For long-period files, data availability will be based upon the date(s) specified in the bi-weekly report of "Digital Seismic Data Transmitted to Datacomputer" via Arpanet message from the Albuquerque Seismological Laboratory. If the desired data were produced prior to the latest transmission date, it is reasonable to attempt to copy it. Because long-period data are normally continuous, i.e., without pre-planned time gaps, the age factor alone is sufficient to determine availability.

For short-period files, determining data availability is a bit more complicated because the data are not recorded continuously. A table of segment availability will be consulted each time a full cycle of task execution has been completed; the results will be used for deciding whether to attempt to retrieve the desired data.

There still remains the possibility that the desired waveform is not on file, even after we have determined that it could be available. In this case, failure to retrieve the waveform will be our clue that it is missing.

A more detailed description of the files and data we will be dealing with in the course of this project is presented in Section 4. A further discussion of the program and the significant features of its key components is given in Section 5.

## 4. Data Description

Below are the actual Datacomputer descriptions of all the types of files involved in the CCA part of the SWF project. These files, each for March 1st 1978, represent a series of day files for their type of data. The day files are grouped under month nodes which are in turn grouped under year nodes using the Datacomputer's hierarchical directory structure. The descriptions include some comments that are stored with the Datalanguage source in the Datacomputer. These appear between "/*" and "*/". Additional commentary appears before the descriptions.

The SRO data files are described first. These are the files from which the CCA SWF-D program will copy data. Next the Event Summary File is described. It will direct the CCA SWF activity and the results of that activity will be posted back into this file. Lastly, the Signal Waveform File is described. It is into this file that the SWF-D program will store data.

<u>SRO Data Files</u> Below are descriptions of the SRO or Seismic Research Observatory (non-array) data files. The long-period data file definition is given first followed by the file definition for short-period data. In each case, all the data for one day appears in one file.

Long-period data are normally continuous, so the long-period file is simply a list of stations each of which is accompanied by a full day of data. The data for a station are organized as a list of 1440 minutes each containing 180 values consisting of one per second for three components.

Short-period data are not recorded continuously so the amount of data for a station on one day varies. Furthermore, due to the way in which data are received from a station on magnetic tapes, a day of data might be split between tapes. This will cause the data to be stored as two chunks for that station and day in the Datacomputer file.

To accommodate the possibility of two pieces of data from a station, the short-period SRO file has provision for twice as many station segments as the long-period file.

Each station segment is a variable number of one-second records consisting of twenty gain-ranged readings for that second. Since the data are not continuous, the DET one-bit flag, available for each second of data, is set for any second where the previous second may be missing.

For verification and searching purposes, explicit date and time fields are included with the data. These appear at the minute-level for long-period data and at the second level for short-period. Most of the lists (or "vectors") in these files have virtual indexes declared (V=I) which provide convenient symbolic access to an item's offset from the beginning of its list, or access to the item given its index. There are also a number of virtual expressions (VE=...) in the files. These provide convenient symbolic access to "data" which does not actually take up any space in the file but can be calculated, when referenced, from constants, list offsets (or virtual indexes), and real data.

```
SDAC.VELANET.NLPF.Y1978.M03.D01 FILE
  LIST(0,3,16),B=32,IA=32,ID=1,II=0 /* size for now is 16 */
      STATION STRUCT,B=32 /* ONE LIST FOR EACH SITE *
          STA STR(5),B=8,I=D /* station id */
          FLAG INT,VE=0
          FILLER BYTE,B=24 /* FILLER */
          STINDEX BYTE,V=I
          DATA LIST(1440)  /* # of minutes per day */
              SAMPLES STRUCT,B=32 /* one minute of data */
                  INDEX BYTE,V=I
                  DATE INT(6),B=8  /* Oyyddd */
                  TIME INT(8),B=8  /* hhmmsscc */
                  PAD BYTE,B=8  /* force 24/32 bit boundary */
                  V INT,VE=1 N INT,VE=2 E INT,VE=3
                      /* use with "type" below */
                  BLOCK STRUCT,B=8 /* Albuquerque supplies */
                      ABSENT BYTE,B=3 /* 1=data absent */
                      OP LIST(3)
                          ERR STRUCT
                              TYPE BYTE,V=I /* v,n,e components */
                              BIT BYTE,B=1 /* 1=operator declared bad */
                          END
                      FILL BYTE,B=2 /* 2 filler bits, undefined */
                  END
                  TIMESERIES LIST(180),B=32
                     SECOND STRUCT
                       SINDEX BYTE,V=I
                       SEC INT,VE=1+(SINDEX-1)/3
                       TYPE INT,VE=SINDEX-((SINDEX-1)/3)*3 /*V,N,E  */
                       DATUM BYTE,B=16
                     END
              END
      END;
```

```
SDAC.VELANET.NSPF.Y1978.M03.D01 FILE /* segmented once per day */
   LIST(0,10,32),B=32,BI=25,IA=56,ID=10,II=0  /* size for now IS 10 */
                    /* up to 32 pieces of days */
      STATION STRUCT,B=32 /* one list for each OF 10 sites */
         STA STR(5),B=8,I=D /* station id */
         STINDEX BYTE,V=I
         COUNT BYTE,B=24  /* count of seconds stored */
         FLAG INT(1),B=8,I=D /* NON-ZERO IF FIRST OF 2 PARTS */
         FILLER BYTE,B=24 /* FILLER */
         DATA LIST(,6700,86400),C=COUNT  /* 6700 SECONDS FOR NOW */
            SAMPLES STRUCT /* once per second */
               INDEX BYTE,V=I
               DATE INT(6),B=8  /* 0yyddd */
               TIME INT(8),B=8  /* hhmmsscc */
               PAD BYTE,B=8  /* force 24/32 bit boundary */
               BLOCK STRUCT,B=8 /* Albuquerque supplies */
                  ABSENT BYTE,B=3  /* 1=data absent */
                  OPERR BYTE,B=1  /* 1=operator declared channel bad */
                  DET BYTE,B=1  /* 1 for first second of detection */
                  FILL BYTE,B=3  /* 3 filler bits, undefined */
               END
               TIMESERIES LIST(20),B=32
                  DATUM BYTE,B=16 /* gain-ranged format */
            END
      END;
```

<u>The Event Summary File</u> is the most complex of the files involved. It consists of a list of events with many parameters associated with each event. Each event also has an inner list of arrivals associated with it. Each "arrival" represents an actual or predicted arrival of a seismic wave front at a station. There are a variable number of arrivals per event and a variable number of events per day file. All of the entries in the event summary file are created by the Seismic Data Analysis Center, Alexandria, Virginia.

Each completed arrival entry contains enough information to locate the actual waveform data, if present, in either a raw data file or the Signal Waveform File for that day. The WAVEFORMAVAIL field for each arrival has a "Y" or "C" if waveform data exists, an "N" if it does not, and a "T" if this is a request for CCA to move waveform data to the Signal Waveform File. The SWF-D program will ultimately change any "T" to a "Y" or "N" as appropriate and in the case of a "Y" (success in moving waveform data) will also set the DATASEGSTART date and time.

The EVENTNUM string shown below will be split into two fields to correspond to the EVENTID structure in the Signal Waveform file.

```
SDAC.VELANET.PESF.Y1978.M03.D01 FILE LIST (,5,100),B=32
EVENT STRUCT,B=32
        INDEX BYTE,V=I
        EVENTNUM STR(9),B=8
        ORIGIN STRUCT
                DATE INT(6),B=8,I=D      /* OYYDDD */
                TIME INT(8),B=8  END     /* HHMMSSCC */
        STANDEVORIGT INT(4),B=8
        NSTA INT(3),B=8
        HYPOCENTER STRUCT
                COMPSOURCE STR(1),B=8,I=D
                EPICENTERCOMP STR(1),B=8,I=D
                LATITUDE STRUCT
                        LAT INT(5),B=8
                                /* XX.XXX DEGREES */

                        HEM STR(1),B=8,I=D  END
                LONGITUDE STRUCT
                        LONG INT(6),B=8
                                /* XXX.XXX DEGREES */
                        HEM STR(1),B=8,I=D  END
                NSTA INT(3),B=8,I=D
                DEPTH STRUCT
                        DEPTH INT(3),B=8   /* XXX KM */
                        STANDEV INT(3),B=8  /* XXX KM */
                        METHOD STR(1),B=8  END
                CONFIDREGION STRUCT
                        SMAJORAXIS INT(5),B=8
                           /* XXXX.X KM */
                        SMINORAXIS INT(5),B=8
                                /* XXXX.X KM */
                        ANGLE INT(4),B=8  END
                                /* XXX.X DEGREES */
                REGION STRUCT
                        GEOCODE STR(3),B=8,I=D
                        SEISNUM STR(2),B=8,I=D  END
                BODYWAVE STRUCT
                        MBMAG INT(3),B=8   /* X.XX */
                        STANDEV INT(3),B=8
                        NSTA INT(3),B=8  END
                SURFACEWAVE STRUCT
                        MSMAG INT(3),B=8  /* X.XX */
                        STANDEV INT(3),B=8
                        NSTA INT(3),B=8  END
```

```
LOCALMAGNITUDE STRUCT
                MLMAG INT(3),B=8
                SOURCE STR(1),B=8
        END
        PARAM STR(20),B=8
        NPHASE INT(3),B=8
        NARR INT(3),B=8
        ARRIVALS LIST (,500,999),C=NARR
                ARRIVAL STRUCT
                        AINDEX BYTE,V=I
                        STA STR(5),B=8,I=I
                        CHANTYPE STR(1),B=8
                        RATE INT(2),B=8,I=I
                        CHANID STR(4),B=8,I=I
                        GAIN STR(1),B=8,I=I
                        COMP STR(1),B=8,I=I
                        ASSOCCONF STR(1),B=8,I=I
                        DIST INT(4),B=8
                                /* XXX.X DEGREES */
                        AZ INT(4),B=8
                           /* STA TO EPI XXX.X DEGREES */
                        DATASEGSTART STRUCT
                                DATE INT(6),B=8,I=I
                                TIME INT(8),B=8
                        END
                        PHASEARR STRUCT
                                DATE INT(6),B=8,I=I
                                TIME INT(8),B=8
                END
                PHASEID STR(6),B=8,I=I
                        PHASENUM INT(2),B=8
                        PHASECODE STR(2),B=8,I=I
                        AMP INT(7),B=8
                                /* XXXXXX.X NM PK - PK */
                        PER INT(3),B=8 /*XX.X SECONDS*/
                        RES INT(5),B=8 /*XX.XX SECONDS*/
                        USAGE STRUCT
                                LOCATION STR(1),B=8,I=I
                                MBUSE STR(1),B=8,I=I
                                MSUSE STR(1),B=8,I=I
                        END
                        WAVEFORMAVAIL STR(1),B=8,I=I
                END
        END
END;
```

<u>The Signal Waveform</u> contains the interesting sections of data extracted from the raw seismic waveform information. Those sections related to array data will be stored directly by the Seismic Data Analysis Center and will be reflected by entries in the Event Summary file related to these actual waveforms. Non-array (SRO) data will be copied into the Signal Waveform File from the SRO data files by the SWF-D program which operates under the direction of marked predicted entries in the Event Summary File.

Because of the length of some of the data segments desired, the maximum dimension of the TIMESERIES list should be increased to 1800. Also, to make retrieval based on event-number easier, EVENTNUM or possibly EVNUM or both should be inverted (I=D). (Inverting the date field would serve no purpose in a file covering only one day.)

```
SDAC.VELANET.PSWF.Y1978.M03.D01 FILE              /* SEGMENTED DAILY */
       LIST (50,200,2200) ,B=32
               /* 50 DETECTIONS PER DAY * 2 ARRIVALS/SITE * 22 SITES */
               SIGNAL STRUCT           /* ONE PER SITE PER PHASE */
                       ARRIVAL STRUCT
                               EVENTID STRUCT     /*EVENT IDENTIFIER*/
                                     EVDATE INT(5),B=8    /*YYDDD */
                                     EVNUM INT(4),B=8   /* NUMBER */
                               EVENTNUM STR(9),VE=EVDATE!EVNUM
                                END
                               INDEX BYTE,V=I
                               STA STR(5),B=8,I=D          /* STATION */
                               CHANTYPE STR(1),B=8
                                       /* A,B,S,I=ADAPTIVE BEAM,BEAM,SUBARRAY BEAM,
                                       INDIVIDUAL INSTRUMENT */
                               RATE INT(2),B=8
                               CHANID STR(4),B=8,I=D
                                       /* BBUU FOR INFINITE VECOCITY BEAMS */
                               GAIN STR(1),B=8  /* H,L */
                               COMP STR(1),B=8
                                       /* Z,N,E,T,R,1,2,3 NUMERICS FOR ALPA */
                               BEAMAZ INT(5),B=8
                                       /*  0-360 DEGREES: XXX.XX DEGREES */
                               BEAMVEL INT(5),B=8      /* XXX.XX DM/SEC */
                               DETECTCONF STR(1),B=8
                                       /* A,B,C, = SIGNAL, S/N ABOUT 1, S/N LESS THAN 1
                                       AND SIGNAL SUSPECT; ANALYST SAYS */
                               CLIPPED STR(1),B=8 /* Y,N, ANALYST SAYS */
                               ARVLPAD STR(1),B=8 /* PAD BYTE */
                       END
```

```
DATASEGMENT STRUCT
        START STRUCT
                DATE INT(6),B=8           /* OYYDDD */
                TIME INT(8),B=8           /* HHHMMSSCC */
        END
        NSAMP INT(4),B=8
                /* DATA SAMPLES IN DATASEGMENT */
        DATAFORMAT STR(1),B=8
                /* I,G: INTEGER, GAINRANGED */
        SCALEFACTOR INT(8),B=8
                /* XXX.XXXXX NANOMETERS PER LEAST SIGNIFICANT BIT */
        DATAPAD STR(1),B=8
        TIMESERIES LIST(,1600),B=16,C=NSAMP
                DATA STRUCT
                DINDEX BYTE,V=I   /*DATA POINT INDEX*/
                DATUM BYTE,B=16
                , - SP: 20 SEC OF NOISE FOLLOWED BY 60 SEC OF SIGNAL
                FOR 20 SAMPLES/SEC DATA.  SP WITH 10 S/S WILL
                USE ONLY 800 16 BIT BYTES.  LP: 300 SEC OF NOISE
                FOLLOWED BY 1300 SEC OF SIGNAL. */
                END
        END
END;
```

# 5. Program Description

This section discusses the significant technical aspects of the key components of the SWF-D program. The factoring of the program description into components is presented here for conceptual clarity rather than as an implementation requirement or constraint.

The overall approach is defined by the sequence of tasks that need to be performed:

Task 1: retrieving ESF requests,

Task 2: determining data availability,

Task 3: copying the desired waveforms into the SWF,

Task 4: updating the ESF, and

Task 5: maintaining a log.

The solutions devised for performing these tasks are described in more detail below. The SWF-Demon program control unit which ties the various parts together is described last.

The Task 1 component will scan all entries in the Event Summary File for arrivals flagged by a "T" in the WAVEFORMAVAIL field. The output of this component will be a Tenex disk file containing a list of the arrivals which will be used by the Task 2 component to check for anticipated data availability.

A starting date will be assembled into the program to mark the earliest event entry in the Event Summary File to be scanned. The program will automatically update it as time progresses and the desired waveforms are copied (or it is determined that they will never be available); and this date, in conjunction with the date of the most recently filed event will bracket the time frame of interest to apply when scanning the ESF. Provision has also been made to adjust the date by programmed interrupt.

Initially, we will process a maximum of 1000 requests at a time on each cycle through the tasks. This number is related to the construction of the ESF daily file which allows for recording up to 1000 arrivals per event. However, the program is not limited in terms of the number of events it can scan, and will certainly continue searching for its ration of requests within the search window.

The program will process the daily ESF files sequentially in chronological order, accumulating the requests, and periodically recording the limits of the search in terms of ESF filename and event-ID. This information will be used for more efficiently restarting the program after crashes.

The following information will be retrieved from the ESF for each request: the event number assigned by NEP, the station-site identifier, phase name, channel name and type, component, rate, gain, and the data segment start date/time field. In addition, file position information (the INDEX and AINDEX fields) will be retrieved. For the precise Datalanguage PORT definition, see Appendix A.

The file definition for the EVENTNUM field is currently described as a 9-byte field of 8-bit bytes, and will need to be changed to conform to the EVENTID structure of the Signal Waveform File.

The Task 2 component will work from the list of requests produced by the Task 1 component. The output of this component will be a chronologically-ordered list of NLPF and NSPF files which will be used by the Task 3 component to locate the desired waveform data.

The problem here is to determine whether it would be worthwhile to make an attempt to locate the waveform data -- worthwhile in terms of there being a high probability of finding the data, rather than in terms of the resources that would be expended to traverse the vast amount of data involved.

Because of the nature of the database (described in detail in Section 4 above), and for any of several causes related to the characteristics of the information-gathering process itself, there remains the likelihood that the desired waveform data is either not yet on file (i.e., has not yet been transmitted to the Datacomputer), or even that it is never going to be available.

It is important to note that the automatic elimination of any candidate waveform on any basis related to seismic significance or validity is not within the scope of this project. On the other hand, there will be requests for data which the program can eliminate automatically based upon known segment availability. Discarded requests will merely be recorded on an operations log for later posting to the ESF by the Task 4 component.

Determining data availability is based, in part, upon advice received via Arpanet messages from J. Hoffman which list, by station, the dates of the most recent SRO data transmitted to the Datacomputer.

On the 22nd of May, for example, the advisory message indicated that all but one station's data had been transmitted for the month of January, and that many were well into February. For the period of operation between the receipt of this message and the next such message, we should pick a cut-off date for our own time frame of sometime early in February, in order to strike a balance between leaving till later a lot of data which could be processed, and uselessly requesting data which have not yet arrived.

To accommodate this operation, the program is designed to accept a manually-input date without requiring recompilation; and a test service period will help us derive the most efficient date to choose.

For long-period files, this date will be our only basis for attempting to locate a selected waveform. The age factor alone is sufficient to determine availability in this case because the NLPF files are continuous.

For short-period files, the program will, in addition, check a table of segment start/stop times.

The absent, tardy or otherwise unavailable waveforms will again be treated as active requests on subsequent cycles through the tasks until the beginning part of the Task 1 component search window moves past the arrival time.

The Task 3 component works from the input list prepared by the Task 2 component to locate and append the selected seismic data to the Signal Waveform File. The output of this component is a fresh list of the waveforms which were successfully copied to the SWF, plus a list of the failures.

A pre-established work schedule will be observed. The time of day to begin work, the number and duration of rest periods, and the amount of work to do during work sessions will have been set externally by the program control unit (described later in this document).

For purposes of evaluation early in the service period, a script of the Datacomputer sessions will be appended to the operations log following each work interval.

The program will be capable of resuming its activities beginning with either the first of a new set of waveforms to copy, or a repeat of an interrupted request.

The Task 4 component will use the list of completed requests which the Task 3 component produces, in conjunction with the list of discarded requests which the Task 2 component produces in the course of checking data availability, to update the WAVEFORMAVAIL field of the Event Summary File.

Eventually, all "T"-marked arrivals will be changed to indicate either:

"Y" - some data was copied, or

"N" - no data is available.

The DATASEGSTART date/time field will also be set for the completed requests.

The Task 5 component will maintain, as its primary function, a log of the SWF-D program activities on a Tenex disk file. This component is effectively a package of utility subroutines which can be called by any of the components as well as from the top-level SWF-Demon control unit.

As each task is begun or completed, or reaches a point in operation from which it would be convenient or efficient to resume execution if interrupted, a state variable will be written onto the log, along with task-dependent information sufficient to reactivate it.

Enough information will be accumulated to produce, by examining the operations log, a summary of program activities (i.e., a list of rejected requests, a count of attempts to locate absent data, a count of processed requests, etc.) for each cycle through the tasks.

The SWF-Demon program control unit contains the logic for initiating or re-initiating program execution. It is responsible for generating the working environment, creating the dynamic program structure, testing the operating system load average and Datacomputer status, examining the operations log for task status information (and creating new log files), setting up and calling the next task to be performed or restarting an interrupted task.

We will allow for changing the periodicity and the load average under which the program will operate by programmed interrupt. The general design philosophy has been to provide an efficiently operating service through programmed flexibility.

## A. Datacomputer PORT Definitions

```
/* PORT FOR EXTRACTING ALL NEEDED INFO ON ARRIVALS */

CREATE SWF.REQ PORT LIST (,9999),P=EOF,B=36
        REQUEST STRUCT
                INDEX BYTE       /*EVENT*/
                EVENTNUM STR(9),B=8
                AINDEX BYTE      /*ARRIVAL*/
                STA STR(5),B=8
                CHANTYPE STR(1),B=8
                RATE STR(2),B=8
                CHANID STR(4),B=8
                GAIN STR(1),B=8
                COMP STR(1),B=8
                DATASEGSTART STRUCT
                        DATE INT(6),B=8
                        TIME INT(8),B=8
                END /*DATASEGSTART*/
                PHASEID STR(6),B=8
        END /*REQUEST*/;
```

```
/* PORT FOR SENDING OVER THE DATA FOR
   UPDATING THE EVENT SUMMARY FILE
   AND APPENDING TO THE SEISMIC WAVEFORM FILE */
CREATE SWF.PUT PORT STRUCT,P=EOF
        ESFPUT LIST(1,999),B=36,C=1
                ESFITEM STRUCT
                        INDEX BYTE
                        AINDEX BYTE
                        DATASEGSTART STRUCT
                                DATE INT(6),B=8
                                TIME INT(8),B=8
                        END /*DATASEGSTART*/
                        WAVEFORMAVAIL STR(1),B=8
                END /*ESFITEM*/
        SWFPUT LIST(,999),B=36,C=1
                SWFITEM STRUCT
                        EVENTID STRUCT
                                EVDATE INT(5),B=8
                                EVNUM INT(4),B=8
                        END /*EVENTID*/
                        STA STR(5),B=8
                        CHANTYPE STR(1),B=8
                        RATE STR(2),B=8
                        CHANID STR(4),B=8
                        GAIN STR(1),B=8
                        COMP STR(1),B=8
                        DATASEGMENT STRUCT
                                START STRUCT
                                        DATE INT(6),B=8
                                        TIME INT(8),B=8
                                END /*START*/
                                SCALEFACTOR INT(8),B=8
                                STATIONI BYTE
                                 /*STATION ENTRY INDEX*/
                                STARTI BYTE
                                  /*DATA START INDEX*/
                                ENDI BYTE /*DATA END " */
                                TYP BYTE /*V,N,E*/
                        END /*DATASEGMENT*/
                END /*SWFITEM*/
        END /*PUT*/;
```

[CCA a]
"Datacomputer Technical Bulletin Number 2, DCSUBR: Functional Specifications", Jerry Farrell, July 1976, Computer Corporation of America, 575 Technology Square, Cambridge, Massachusetts 02139.

[CCA b]
"Datacomputer Technical Bulletin Number 8, The CCA Datacomputer Status Server", Donald E. Eastlake, III, April 1978, Computer Corporation of America, 575 Technology Square, Cambridge, Massachusetts 02139.

[CCA c]
"Datacomputer Version 5 User Manual", July 1978, Computer Corporation of America, 575 Technology Square, Cambridge, Massachusetts 02139.

[CCA d]
"CCA-77-10 Very Large Databases: Final Technical Report", Robert H. Dorin and Joanne Z. Sattley, August 30, 1977, Computer Corporation of America, 575 Technology Square, Cambridge, Massachusetts 02139.

[CCA e]
"CCA-77-11 Tertiary Memory Access and Performance in the Datacomputer", Donald E. Eastlake, III, June 30, 1977, Computer Corporation of America, 575 Technology Square, Cambridge, Massachusetts 02139.

[CCA f]
"CCA-78-04 Datacomputer and SIP Operations: Final Technical Report", January 1978, Donald E. Eastlake, III, (Project Manager), Robert H. Dorin, Jerry Farrell, Stephen A. Fox, David Kramlich, Matthew Maltzman, Gerald E. Maple, James Schmolze and Steven A. Zimmerman, Computer Corporation of America, 575 Technology Square, Cambridge, Massachusetts 02139.

[BBN]
"BCPL Manual", September 1974, Bolt, Beranek and Newman, Inc., 50 Moulton Street, Cambridge, Mass. 02138.

[TELEDYNE]
VELANET ESF/SWF PROCESSING, Joseph Greenhalgh, Design Draft, February 1, 1978, Teledyne Geotech, Alexandria Laboratories, P. O. Box 334, Alexandria, Virginia 22313.