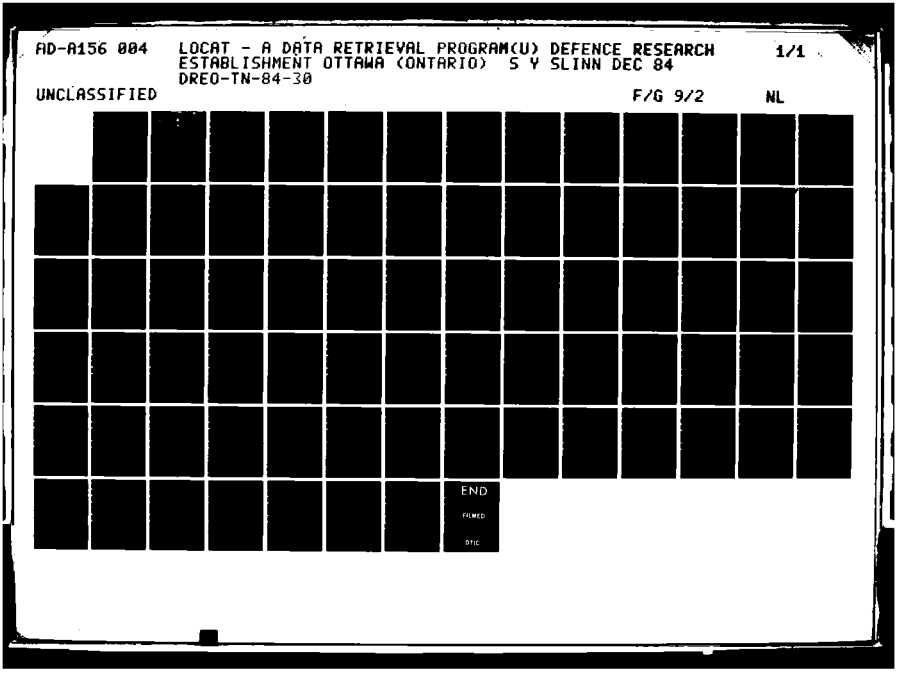
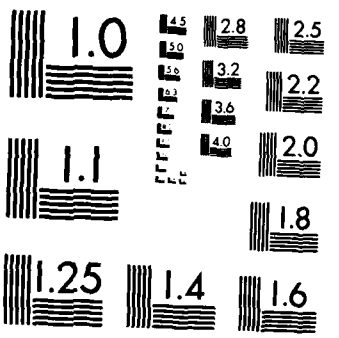

AD-A156 004 LOCAT - A DATA RETRIEVAL PROGRAM(U) DEFENCE RESEARCH ESTABLISHMENT OTTAWA (ONTARIO) S Y SLINN DEC 84 DREO-TN-84-30
1/1
UNCLASSIFIED
F/G 9/2
NL
END
FILMED
DTIC



National Defence / Défense nationale

LOCAT - A DATA RETRIEVAL PROGRAM

by

Suzanne Y. Slinn

DEFENCE RESEARCH ESTABLISHMENT OTTAWA
TECHNICAL NOTE 84-30

Canada

December 1984
Ottawa

National Defence / Défense nationale

LOCAT - A DATA RETRIEVAL PROGRAM

by

Suzanne Y. Slinn
SARSAT Project Office
Electronics Division

DEFENCE RESEARCH ESTABLISHMENT OTTAWA
TECHNICAL NOTE 84-30

PCN
33X00

December 1984
Ottawa

ABSTRACT

The computer program, LOCAT, a data retrieval package for the SARSAT Evaluation Facility database, is described for use on an HP-1000 computer. The program provides the user with flexibility over what data is to be retrieved from the database. The following input options are available:

- satellite(s)
- frequency(s)
- location and radius
- date range

The data retrieved is presented in the form of four outputs, each of which provides different information.

LOCAT is documented in terms of a brief description of the package, its capabilities, a guide on how to use it, and how to compile and load it. The source code for the routines written is provided in the Appendices.

RÉSUMÉ

Le présent résume décrit le programme de base de données LOCAT utilisé conjointement avec l'ordinateur HP-1000 pour traiter les données recueillies pendant la phase d'évaluation du système SARSAT. Ce programme donne à l'usager toute la souplesse voulue pour recouvrer les données dont il a besoin. Ainsi, celles-ci peuvent être rappelées d'après les paramètres suivants:

- satellite(s)
- fréquence(s)
- lieu et rayon
- étendue dans le temps

L'usager a également la possibilité de choisir parmi quatre modes de présentation différents.

Le résumé comporte une brève description du programme LOCAT et de ses possibilités, le mode d'emploi ainsi que le protocole de compilation et de chargement. Le code source est fourni en annexe.

TABLE OF CONTENTS

LIST OF FIGURES

A1

1.0 INTRODUCTION

LOCAT is a program used to facilitate the retrieval of data from the SARSAT Evaluation Facility (SEF) database. It is written using FORTRAN 4X and IMAGE/1000, and is intended to be used on an HP-1000 computer with a RTE IVB operating system.

The SEF database was established to process Canadian SARSAT evaluation data collected from various sources including the SARSAT Local User Terminal (LUT), the Canadian Mission Control Centre (CMCC) located at CFB Trenton and the Rescue Coordination Centres (RCC) located across Canada. Because of limitations in the SEF software, which was developed under contract, and in order to address specific evaluation requirements, it was necessary to build a package of software routines to support SARSAT studies. LOCAT was the first of a series of analytical software programs to meet these requirements.

This report documents the program LOCAT in terms of the program overview which summarizes the data retrieval criteria and the output files produced, a user's guide which provides instructions on how to run the program, and finally, information on compiling and loading the program in the SEF operating environment. Appendix A contains a sample run and Appendix B documents the LOCAT source code.

2.0 PROGRAM OVERVIEW

The SEF database is a time ordered structure containing SARSAT evaluation data from a variety of sources. During operations, SARSAT facilities detect transmissions from Emergency Locator Transmitters (ELTs) or their marine equivalent, the Emergency Position Indicating Radio Beacon (EPIRB), which may have been activated because of an air or marine distress incident. The SARSAT facilities, specifically the ground tracking stations, through Doppler processing, derive an estimate location of the beacon which could be transmitting at 121.5 and/or 243 MHz or 406 MHz. These data are then passed on to the operational Search and Rescue community for actioning. During each step of this sequence of events, data

is collected in order to evaluate SARSAT performance. Evaluation data consists of the estimated locations, technical signal processing parameters developed as a result of the Doppler process, and operational SAR data.

The LOCAT program provides the convenient capability to retrieve SARSAT data - which could cover a period of up to six months - in terms of:

- Time or time interval of interest;
- Satellite or combination of satellites used;
- Location of interest;
- Beacon frequency.

The program output consists of data files available for subsequent processing, or data listing. The level of parameter detail on output is under user control.

The LOCAT design makes use of the soft key facilities provided by the HP 2648A graphics terminal. This design philosophy is described and then the soft key displays are discussed. As noted previously, the level of output is under user control. The content of the output data files are defined.

2.1 DESIGN PHILOSOPHY

The LOCAT program was designed to enable the retrieval of data according to user specified criteria and to be as self-explanatory as possible for ease of use. The program is modular in structure for ease of modification. Figure 1 illustrates the calling sequence hierarchy.

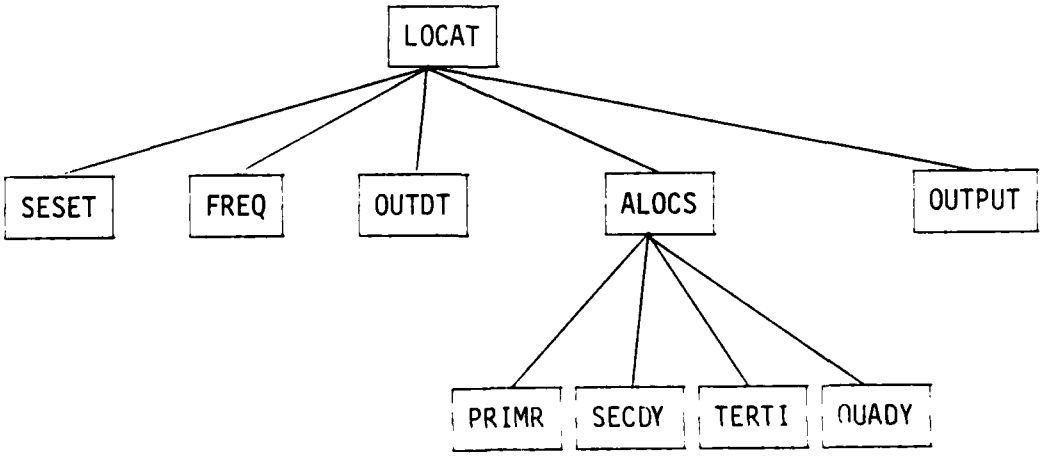


Figure 1: LOCAT - Calling Sequence.

The functions of each of these routines is as follows:

- LOCAT - main line program, request SAT/LOC data definition.
- SESET - requests query time definition and logical unit for the output destination.
- OUTDT - requests output definition.
- ALOCS - interrogates database, gets the data for output.
- OUTPT - produces the output listings.
- FREQ - requests frequency definition.
- PRIMR - gets the data for the primary output.
- SECDY - gets the data for the second output.
- TERTI - gets the data for the third output.
- QUADY - gets the data for the fourth output.

The HP 2648A terminal soft keys have been used extensively in the program in order to avoid the situation of the user having to answer a multitude of questions to determine the retrieval criteria.

When a new set of soft keys is displayed, a prompt appears on the console requesting the user to specify the choice of the presently displayed retrieval criteria. This is done by pressing the soft key corresponding to the user's choice. An '*' will appear on the screen beside the displayed soft key to indicate that the key was pressed. At this stage the query can be changed. This is done by simply pressing the soft key again. The '*' disappears indicating that the choice has been discarded.

2.2 SATELLITE AND LOCATION SOFT KEY DISPLAY

The soft keys that are displayed to allow the user to choose the desired satellite(s) and location are illustrated in Figure 2. Should none of the soft keys be selected, the default values are "all" satellites and "all" locations.

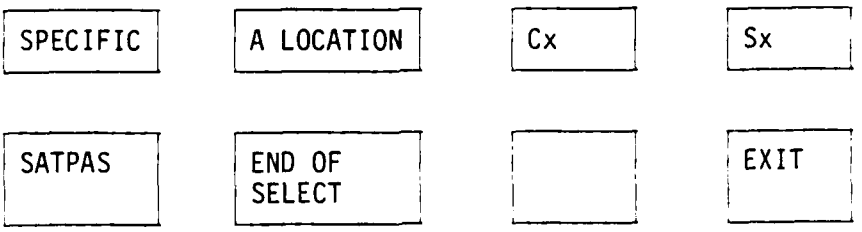


Figure 2: SAT/LOC Soft Key Display

Should data be required for a specific satellite(s), the SPECIFIC SAT soft key is pressed. Once the display prompting for satellite data is obtained, the user inputs the desired satellite name in the form S1, C1, etc. A maximum of five satellites can be entered, with each name separated by a comma or space.

If the query involves obtaining data within a certain radius of a specified location, the LOCATION soft key is pressed. The user is then prompted to enter the longitude, latitude of the location, the radius desired, and the region and case number of this location. If this latter information is not available, the carriage return is pressed.

If data is desired for only the COSPAS satellites, utilize the Cx soft key. Data for all the COSPAS satellites will then be retrieved.

Similarly, pressing the Sx soft key causes the data for all the SARSAT satellites to be retrieved.

Data for a specific satellite pass can be retrieved by pressing the SATPAS key. The user is then prompted to input the specific satellite pass number.

To indicate to the program that the user is finished with the present selection, the END OF SELECT key is pressed and the program continues on.

The EXIT key, available with each display, allows the user to exit the program.

2.3 FREQUENCY SOFT KEY DISPLAY

The soft keys that are displayed to allow the user specification of the desired frequency are illustrated in Figure 3.

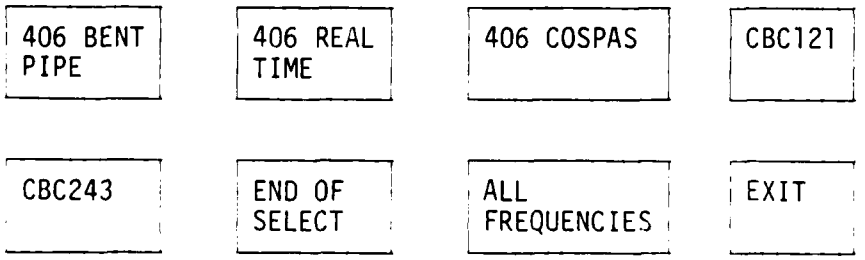


Figure 3: Frequency Soft Key Display

The soft key definitions are:

- 406 Bent Pipe - unprocessed 406 MHz data from the Search and Rescue repeater.
- 406 Real Time - 406 MHz data that is processed onboard the satellite and provided to the LUT on the 2.4 kilobit downlink.
- 406 COSPAS - stored memory dump 406 MHz data from the satellite.
- CBC121 - 121.5 MHz data.
- CBC243 - 243 MHz data.
- All Frequencies - data at all frequencies are retrieved.

Only the following combinations of frequencies are permissible:

- all frequencies;
- 406 Bent Pipe;
- 406 Real Time;
- 406 COSPAS;
- CBC121;
- CBC243;
- CBC121 and CBC243.

2.4 OUTPUT SOFT KEY DISPLAY

The soft keys that are displayed to allow user specification of the desired output are illustrated in Figure 4. There is a choice of four different outputs available, each retrieving different data from the database. Any combination of these outputs is permissible. The definition of these various outputs are given in subsequent sections.

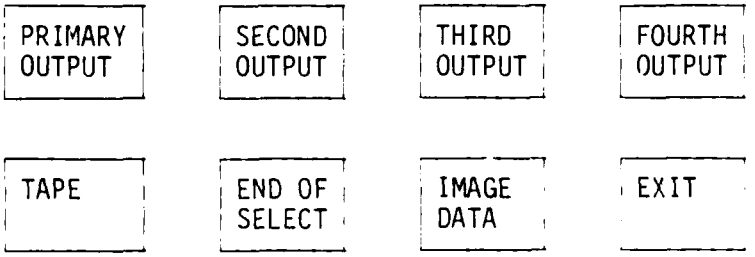


Figure 4: Output Soft Key Display.

The data files produced by the program can be utilized in subsequent processing by pressing the TAPE soft key. This causes the disc file(s) to be stored onto magnetic tape.

Each ELT location calculated by the LUT also has a Doppler image solution. LOCAT normally retrieves data for the true ELT location. The IMAGE DATA soft key enables the retrieval of data for the image solutions as well as for the true solutions.

2.4.1 Primary Output File

The primary output file option displays data with the following header:

```
                                          PRIMARY   SECONDARY
DATA  SATPAS  MCCREF  EVENT  MESSNT  LAT  LONG  LAT  LONG  DIFF  DLAT DLONG

LOCATION LATITUDE - 45.0000   LONGITUDE - 75.0000   RADIUS - 100.0
```

The output data definition is as follows:

DATE - year, month, day of acquisition of signal.

SATPAS - satellite identification and pass number.

MCCREF - the CMCC reference number.

EVENT - the LUT assigned reference number.

MESSNT - flag indicating whether the ELT data was sent via an alert message to the CMCC (=0,no =1,yes).

LAT - calculated latitude of ELT location expressed in degrees.

LONG - calculated longitude of ELT location expressed in degrees.

SECONDARY LAT, LONG - calculated latitude and longitude of the image ELT location expressed in degrees.

DIFF - the distance in kilometres between the primary LAT and LONG and the user specified latitude and longitude.

DLAT - the distance in kilometres between the primary LAT and user specified latitude.

DLONG - the distance in kilometres between the primary LONG and the user specified longitude.

LOCATION LATITUDE, LONGITUDE AND RADIUS - all retrieved data are to be within the radius of this location. If the user does not specify a location and radius, DIFF, DLAT, DLONG and the location latitude, longitude and radius are displayed as zero.

2.4.2 Second Output File

Depending on the frequency chosen for the data to be retrieved, 121.5/243 or 406 MHz, two different outputs are displayed for this output option.

The following output describes 121.5/243 MHz data.

SECOND OUTPUT

PRIMARY LOCATION

CTA POINTS SDEV TREND QUAL PROB NMWLS TCA QTIME LOSTIM BIAS CORR SCORE

PRIMARY LOCATION - indicates that all the data in the second output file is for the primary location found in the primary output file.

CTA - cross track angle in degrees.

POINTS - number of frequency measurements.

SDEV - standard deviation of residuals in Hz.

TREND - trend factor of residuals in Hz.

QUAL - quality factor of CBC data (sum amps).

PROB - probability of true solution, as a percentage.

NMWLS - number of WLS iterations.

TCA - time of closest approach, in hours from the date of acquisition of signal (AOS).

QTIME - time of AOS in hours from midnight of the DATE.

LOSTIM - date of loss of signal (LOS), in hours from date of AOS.

BIAS - ELT frequency bias, expressed in Hz.

CORR SCORE - ELT identifier consisting of four ASCII blanks and a real correlation score.

The following output describes 406 MHz data. Only the headings that are different from the previous description will be expanded upon.

SECOND OUTPUT

PRIMARY LOCATION

CTA POINTS SDEV TREND ELTANG PROB NMWLS TCA QTIME LOSTIM BIAS ELT ID

ELTANG - elevation angle between the ELT and the satellite.

ELT ID - ELT identifier stating the country, user and beacon identification code.

2.4.3 Third Output File

THIRD OUTPUT

PRIMARY LOCATION

DRIFT CTAI TCAI MAJAX MINAX AMEAN BIASI

PRIMARY LOCATION - indicates that all the data in the third output file is for the primary location found in the primary output file.

DRIFT - ELT frequency drift, in Hz/min.

CTAI - initial estimate of CTA in degrees.

TCAI - initial estimate of TCA in seconds.

MAJAX - major axis of error ellipse, in km.

MINAX - minor axis of error ellipse, in km.

AMEAN - average of data residuals in Hz.

BIASI - initial estimate of BIAS in Hz.

2.4.4 Fourth Output File

FOURTH OUTPUT

PRIMARY LOCATION

VARCTA VARTCA VARBIA VARDI CORCT CORCB CORCD CORTB CORTD CORBD

PRIMARY LOCATION - indicates that all the data in the fourth output file is for the primary location found in the primary output file.

```
C        check if the four' output file is to be out on tape
         IF (FOURTH .EQ. 7HEM
           IBUF1(I)=2HIF
           IBUF1(I+1)=2HOU
           IBUF1(I+2)=2HRT
           IBUF1(I+3)=2HH,
           I=I+4
         ENDIF
         ENDIF
C        terminate the file selection with a /E in the buffer
         IBUF1(I)=2H/E
         J=2*(I)
C        send the transfer file command to be operating system
         CALL EXEC(14,2,IBUF1,J)
       ENDIF
       WRITE(14,1010)POSN,LOCLAT,LOCLNG,RADIUS,REGION,CASENO
      ENDIF
C
  900 CONTINUE
C     close the database
      CALL DQCLS(IBASE,IDLUT,1,ISTAT)
      IF (ISTAT(1) .EQ. 0) GO TO 999
      WRITE(LU,1005) ISTAT(1)
C
  999 CONTINUE
      ENDFILE UNIT
C     close all the files used
      CLOSE (UNIT)
      CLOSE(10)
      CLOSE(11)
      CLOSE(12)
      CLOSE(13)
      CLOSE(14)
      CLOSE(21)
      CLOSE(22)
      CLOSE(23)
      CLOSE(24)
 9999 CALL EXEC(6,0,0)
C
C          FORMATS
C
 1000 FORMAT ("Enter target location data:")
 1001 FORMAT ("Specify the search location")
 1003 FORMAT ("Which data do you wish?  ")
 1004 FORMAT ("**ERROR** RADIUS specified incorrect")
 1005 FORMAT ("IMAGE FILE ERROR (SELOC) - ",I5)
 1006 FORMAT ("    ")
 1007 FORMAT (" LOCATION LATITUDE - ",F10.4,2X,"LONGITUDE - ",
     -        F10.4,2X,"RADIUS - ",F9.1)
 1008 FORMAT("Enter up to 5 specific satellites")
 1009 FORMAT(5(A2,1X))
 1010 FORMAT(I4,1X,F10.4,1X,F10.4,1X,F9.1,1X,A2,1X,I4)
 1011 FORMAT(' Do you wish another query?')
 1012 FORMAT(A1)
 1013 FORMAT(' Enter SATPAS desired  ie: C1 05407  ')
 1014 FORMAT(4A2)
 1015 FORMAT(' Enter the name of the database to be used')
 1016 FORMAT(5A2)
      END
```

```
      CLOSE(30)
C
C     read the LUTELF records and determine if they meet the desired criteria
C
C     CALL ALOCS
      CALL EXEC(9,6HALOCS )
      WRITE(LU,1011)
      READ(LU,1012)ANSWER
C
      OPEN(30,FILE='SCRATCH',STATUS='OLD')
      READ(30,1023)POSN,START
 1023 FORMAT(I5,1X,I5)
      CLOSE(30)
C
      IF ((POSN - START) .NE. 0) FOUND = .TRUE.
C
      IF (ANSWER .EQ. 2HY ) GO TO 5
C
      IF (FOUND) CALL OUTP2
      IF (FOUND) CALL OUTDT
C
C     store the output files onto tape if desired
C
      IF (FOUND) THEN
      IF (TAPE) THEN
         I=7
         IBUF1(I)=2HHE
         IBUF1(I+1)=2HAD
         IBUF1(I+2)=2HR,
         I=I+3
C        check if the primary output file is to be put on tape
         IF (PRIM) THEN
           IBUF1(I)=2HPR
           IBUF1(I+1)=2HIM
           IBUF1(I+2)=2HAR
           IBUF1(I+3)=2HY,
           I=I+4
         ENDIF
C        check if the second output file is to be put on tape
         IF (SECOND) THEN
           IBUF1(I)=2HSE
           IBUF1(I+1)=2HCO
           IBUF1(I+2)=2HND
           IBUF1(I+3)=2H,
           I=I+4
         ENDIF
C        check if the third output file is to be put on tape
         IF (THIRD) THEN
           IBUF1(I)=2HTH
           IBUF1(I+1)=2HIR
           IBUF1(I+2)=2HD,
           I=I+3
         ENDIF
C        check if the fourth output file is to be put on tape
         IF (FOURTH) THEN
           IBUF1(I)=2HFO
           IBUF1(I+1)=2HUR
           IBUF1(I+2)=2HTH
           IBUF1(I+3)=2H,
           I=I+4
         ENDIF
      IF (IMAGE) THEN
C        check if the primary output file is to be put on tape
         IF (PRIM) THEN
           IBUF1(I)=2HIP
           IBUF1(I+1)=2HRI
           IBUF1(I+2)=2HMA
           IBUF1(I+3)=2HRY
           IBUF1(I+4)=2H,
           I=I+5
         ENDIF
C        check if the second output file is to be put on tape
         IF (SECOND) THEN
           IBUF1(I)=2HIS
           IBUF1(I+1)=2HEC
           IBUF1(I+2)=2HON
           IBUF1(I+3)=2HD,
           I=I+4
         ENDIF
C        check if the third output file is to be put on tape
         IF (THIRD) THEN
           IBUF1(I)=2HIT
           IBUF1(I+1)=2HHI
           IBUF1(I+2)=2HRD
           IBUF1(I+3)=2H,
           I=I+4
         ENDIF
```

```
C
C     clear the '8' from the select array
C
      DO 4 RK=1,8
        KEYS(1,RK)=2H
 4    CONTINUE
C
 210 CONTINUE
C
C     if specific satellites wanted, read the desired satellites
C
      DO 114 I=1,5
 114    SPESAT(I)=2H
C
      IF (SAT) THEN
        WRITE(LU,1008)
        READ(LU,1009)(SPESAT(I),I=1,5)
      ENDIF
C
C     read desired lat,long, radius if not all the locations are desired
C
      IF ( NOT (ALLUT)) THEN
        WRITE (LU,1000)
        WRITE (LU,1001)
        CALL RETLL (LOCLAT,LOCLNG,FLG)
        IF (FLG .EQ. 1) GO TO 9999
C
C
   50   RADIUS = 10.
        CALL CLONS(6HPRESP ,LU,SYS,
     -             12H,,1,I,Radius,-12,IPARM)
        IF (IPARM(1) .NE. 0)  RADIUS = IPARM(1)
      WRITE(1,554)
 554  FORMAT(' ENTER THE REGION AND CASE NUMBER OF THIS LOCATION.')
      WRITE(1,557)
 557  FORMAT(' THE CODES FOR THE REGIONS ARE:  HZ - HALIFAX,',/,
     1 'TR -TRENTON, ED - EDMONTON, YJ -VICTORIA. ')
      WRITE(1,556)
 556  FORMAT(' ENTER THEM ACCORDING TO THIS EXAMPLE:  TR 1111')
      READ(1,555)REGION,CASENO
 555  FORMAT(A2,1X,I4)
      ENDIF
C
C     determine if specific SATPAS  is desired
C
      IF (SPECIF) THEN
        WRITE(LU,1013)
        READ(LU,1014)(PASS(I),I=1,4)
      ENDIF
C
C     determine which of the four outputs are desired
C
      IF (POSN .EQ. 1) CALL OUTPT
      IF (POSN .NE. 1) THEN
       OPEN(14,FILE='HEADR',IOSTAT=IOS,STATUS='OLD')
       IF (IOS .NE. 0) WRITE(LU,2018)IOS
       IF (IOS .NE. 0 ) GO TO 999
 2018  FORMAT(' ERROR IN OPENING HEADR FILE ',I4)
       DO 65 I1=1,(PLACE-1)
        READ(14,1017)I
 1017   FORMAT(A2)
  65   CONTINUE
      ENDIF
C
      IF (EXIT) GO TO 900
C     determine which of the frequencys are desired
      CALL FREQ
      IF (EXIT) GO TO 900
C
C     display lat, long, radius
C
      WRITE(14,1010)POSN,LOCLAT,LOCLNG,RADIUS,REGION,CASENO
      PLACE=PLACE+1
      START=POSN
      CLOSE(14)
C
      IF (IFBRK(AA) .LT. 0) GO TO 999
C
C     Write the common variables to a scratch file
C
      OPEN(30,FILE='SCRATCH',STATUS='OLD')
      WRITE(30,1018)ALLUT,ALSAT,SAT,SX,CX,(SPESAT(I),I=1,5),PRIM,
     1  SECOND, THIRD,FOURTH
 1018 FORMAT(5(L1,1X),5A2,1X,4(L1,1X))
      WRITE(30,1019)PST,SST,TST,FST,UNIT,DIF,BENT,REALT,COS4,CBC12,
     1  CBC24,ALLFR,POSN,START,EXIT,TAPE,SPECIF,SEVENT,
     1  (PASS(I),I=1,4)
 1019 FORMAT(5(I5,1X),F10.3,1X,6(L1,1X),2(I5,1X),3(L1,1X),I5,1X,4A2)
      WRITE(30,1020)(LODAT(I),I=1,3),(HIDAT(I),I=1,3),STDAT,ENDAT,
     1    (OUTLU(I),I=1,5),RANGE,LOCLAT,LOCLNG,RADIUS,IMAGE,REGION,
     1     CASENO,PLACE,(IBASE(I),I=1,5)
 1020 FORMAT(3A2,1X,3A2,1X,F10.3,1X,F10.3,1X,5A2,1X,I8,1X,3(F10.3,1X),
     1 L1,A2,1X,I4,1X,I3,1X,5A2)
C
```

```
C                EXECUTABLE  CODE
C
C*****************************************************************************
C
      LU=1
      DO 1 I=1,5
 1      IBASE(I)=2H
      WRITE(LU,1015)
      READ(LU,1016)(IBASE(I),I=1,5)
      ISYS = LOGLU(SYS)
      POSN = 1
      FOUND = .FALSE.
      EXIT = .FALSE.
      TAPE = .FALSE.
      REGION = 2H
      CASENO = 0
      PLACE=1
C
C        initialize variables
C
  5   CONTINUE
C
      ALLUT= TRUE.
      ALSAT= TRUE.
      SAT= FALSE.
      CX= FALSE.
      SX= FALSE.
      SPECIF= FALSE.
C
      UNIT = 99
      LUT = .FALSE.
      MCC = .FALSE.
      SAR = .FALSE.
      TEST= .FALSE.
      ALL = .FALSE.
      LOCLAT = 0.0
      LOCLNG = 0.0
      RADIUS = 0.0
C
C        get other input variables
C
      CALL SESET(UNIT,RET)
      IF (RET .NE. 0)  GO TO 999

C
C       search files specified
C
  100 CONTINUE
      CALL FUNKY (KEYS,PROM)
      WRITE (LU,1003)
  101 CALL REIO(1,LU,RK,-2)
      RK = RK -2Hf0
      IF (RK .LT. 1 .OR. RK .GT. 8) GO TO 101
      IF (RK .EQ. 8) THEN
        EXIT= TRUE.
        GO TO 900
      ENDIF
      IF (RK .EQ. 7) GO TO 101
C
C      set $ in key selected and check for further selects
C
      IF (KEYS(1,RK) .EQ. 2H  ) THEN
          KEYS(1,RK) = 2H$
          IF (RK .EQ. 1)     SAT = .TRUE.
          IF (RK .EQ. 2)     ALLUT = .FALSE.
          IF (RK .EQ. 3)  CX = .TRUE.
          IF (RK .EQ. 4) SX = .TRUE.
          IF (RK .EQ. 5) SPECIF = .TRUE.
          GO TO 102
        ENDIF
C
      IF (KEYS(1,RK) .EQ. 2H$ ) THEN
          KEYS(1,RK) = 2H
          IF (RK .EQ. 1)     SAT = .FALSE.
          IF (RK .EQ. 2)     ALLUT = .TRUE.
          IF (RK .EQ. 3)  CX = .FALSE.
          IF (RK .EQ. 4) SX = .FALSE.
          IF (RK .EQ. 5) SPECIF = .FALSE.
        ENDIF
C
      IF (.NOT (SAT .AND. CX .AND. SX)) ALSAT=.TRUE.
C
  102 CALL FUNKY(KEYS(1,RK),RK)
      IF (RK .EQ. 6)  GO TO 200
      GO TO 101
C
  200 CONTINUE
```

```
      REAL LOCLAT, LOCLNG, RADIUS
C
C         LOCLAT - latitude of specified location
C         LOCLNG - longitude of specified location
C         RADIUS - maximum distance between specified location and
C                  file locations
C
      COMMON /LOCN/ LOCLAT, LOCLNG, RADIUS
C
      INTEGER IBASE(5), ILEVL(3), IMODE, ISTAT(10), LIST
      INTEGER ITDTE(3), ITTST(3), ITSAR(3), ITMCC(3), ITSPD(3)
      INTEGER IDLUT(3), IDTST(3), IDSAR(3), IDMCC(3)
C
      INTEGER SBUF(68), LBUF(98), MBUF(76), FBUF(50)
C
      COMMON /BASE/ IBASE, ILEVL, IMODE, ISTAT, LIST, ITDTE,
     -              ITTST, ITSAR, ITMCC, ITSPD, IDLUT, IDMCC,
     -              IDSAR, IDTST, SBUF, LBUF, MBUF, FBUF
C
      DOUBLE PRECISION QTIME, SDT
C
C         SDT    - start date
C
C
C         LODAT  - ascii start date YYMMDD
C         STDAT  - low date in seconds from 1980
C         ENDAT  - high date in seconds from 1980
C
C         OUTLU  - output device, LU or filename
C
      INTEGER RBUF(40), IPARM(5), PBUF(33)
      INTEGER LU, RET
      INTEGER SYS, ISYS, LOGLU
C
C         RBUF   - receiving buffer
C         IPARM  - return buffer for RMPAR call
C         PBUF   - return buffer from PARSE
C
      REAL ONEDAY
      INTEGER DATCH(3), LOOP, DUM(7), I1
      INTEGER UNIT
C
C
      LOGICAL LUT, MCC, SAR, TEST, ALL
C
C         LUT    - flag to indicate LUT location detail file selected
C         MCC    - flag to indicate MCC location detail file selected
C         SAR    - flag to indicate SARSTAT incident detail file selected
C         TEST   - flag to indicate FIELD test master file selectedlected
C         ALL    - all files will be searched
C
      INTEGER KEYS(8,8), PROM(8), RK
      INTEGER FLG, AA, IFBRK
C
C       FLG   - flag to indicate if /E used
C       AA    - dummy parameter for IFBRK
C
C
      COMMON/SELCT/ALLUT,ALSAT,SAT,SX,CX,SPESAT,PRIM,SECOND,THIRD,
     1       FOURTH,PST,SST,TST,FST,UNIT,DIF,BENT,REALT,COS4,CBC12,
     1       CBC24,ALLFR,POSN,START,EXIT,TAPE,SPECIF,SEVENT,PASS,
     1       IMAGE,CONTRY,IUSER,TYPE,ELTANG
      INTEGER CONTRY(3),IUSER,TYPE(4)
      REAL ELTANG
      LOGICAL ALLUT,ALSAT,SAT,SX,CX,PRIM,SECOND,THIRD,FOURTH,SPECIF
      LOGICAL BENT,REALT,COS4,CBC12,CBC24,ALLFR,FOUND,EXIT,TAPE
      LOGICAL IMAGE
      INTEGER SPESAT(5),I,PST,SST,TST,FST,POSN,START,ANSWER
      INTEGER IBUF1(70),J,SEVENT,PASS(4),REGION,CASENO
      INTEGER PLACE
      REAL DIF
C
C       ALLUT -flag to indicate if all locations wanted
C       ALSAT -flag to indicate if all satellites wanted
C
C       SAT   -flag to indicate if specific satellites wanted
C       SX    -flag to indicate if all SARSAT satellites wanted
C       CX    -flag to indicate if all COSPAS satellites wanted
C       SPESAT-array that contains the specific satellites desired
C
      DATA KEYS /'  Specific SAT      A location  ',
     -           '       Cx             Sx      ',
     -           '  Satpas          END of select ',
     -           '                     EXIT       '/
C
      DATA PROM /'f1f2f3f4f5f6f7f8'/
C       KEYS    - array containing contents of soft keys
C       PROM    - array of values to return for soft keys
C       RK      - actual value returned
C
C
      DATA IBUF1/' TR,STPRIM, '/
C
C*****************************************************************************
```

```
      DATA ILEVL /6HENTRY /
      DATA LIST  /040040B/
      DATA ITDTE /6HDATE  /
      DATA ITTST /6HTEST  /
      DATA ITSAR /6HSARNBR/
      DATA ITMCC /6HMCCREF/
      DATA ITSPD /6HSATPAS/
      DATA IDTST /6HFIELD /
      DATA IDSAR /6HSARIF /
      DATA IDMCC /6HMCCELF/
      DATA IDLUT /6HLUTELF/
      END



FTN4
$FILES(15,15)
      PROGRAM LOCAT (3,99)
     -            SEF-40-08  V00  (840726.0907)
      IMPLICIT NONE
C
C-----------------------------------------------------------------------*
C                                                                       *
C     SARSAT EVALUATION FACILITY                                        *
C                                                                       *
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
C                                                                       *
C   DATE    VERSION          DESCRIPTION              AUTHOR            *
C                                                                       *
C 83/10/05    00              - - - - -             SUZANNE Y. SLINN    *
C                                                                       *
C                                                                       *
C                                                                       *
C-----------------------------------------------------------------------*
C DESCRIPTION:                                                          *
C     This program will retrieve data in a certain area    during     *
C     a specified time period                                           *
C     This program will compute distances between locations, and        *
C     produce up to four output files depending on what the user        *
C     specifies.                                                        *
C                                                                       *
C CALLING SEQUENCE:                                                     *
C                                                                       *
C     :RU,LOCAT                                                         *
C                                                                       *
C INPUTS:                                                               *
C                                                                       *
C                                                                       *
C OUTPUTS:                                                              *
C                                                                       *
C EXTERNAL REFERENCES:                                                  *
C                                                                       *
C      Subroutines:                                                     *
C         System:                                                       *
C            RMPAR                                                      *
C            PARSE                                                      *
C            EXEC                                                       *
C            IFBRK                                                      *
C        User:                                                          *
C          MOVBS - move a byte string                                   *
C          SESET - to query the operator for the date range,            *
C                  the device for the output file                       *
C          FUNKY - to set soft keys  (XSCRN in XSEFLB)                  *
C          RETLL - query operator for latitude and longitude            *
C          DQOPN - to open the database                                 *
C          DQCLS - to close the database                                *
C                                                                       *
C                                                                       *
C      Functions:                                                       *
C          GETCH - to get a character from a string                     *
C          NUMBR - to convert ASCII string to integer #                 *
C          PTIME - to convert integer times to double precision seconds *
C                  from 1980                                            *
C                                                                       *
C-----------------------------------------------------------------------*
C
C
C-----------------------------------------------------------------------*
C                                                                       *
C                        DATA DECLARATIONS                              *
C                                                                       *
C-----------------------------------------------------------------------*
C
C
C
      COMMON /LOCC/ LODAT(3), HIDAT(3), STDAT, ENDAT,OUTLU(5),RANGE
      INTEGER IOS
      INTEGER LODAT, HIDAT, OUTLU, RANGE
      DOUBLE PRECISION STDAT, ENDAT
C
```

```
FTN4
      BLOCK DATA
C-----------------------------------------------------------------
C          This is the block common for the program LOCAT
C-----------------------------------------------------------------
C
      INTEGER LODAT, HIDAT, OUTLU, RANGE
C
C        LODAT  - start date in ascii YYMMDD
C        HIDAT  - end date in ascii YYMMDD
C        OUTLU  - output device or file name
C        RANGE  - # of days to be processed
C
      DOUBLE PRECISION STDAT, ENDAT
C
C        STDAT  - start date in double precision seconds from 1980
C        ENDAT  - end date in double precision seconds from 1980
C
      COMMON /LOGG/ LODAT(3), HIDAT(3), STDAT, ENDAT, OUTLU(5),RANGE
C
      REAL LOCLAT, LOCLNG, RADIUS
C
C        LOCLAT - latitude of location specified
C        LOCLNG - longitude of location specified
C        RADIUS - maximum separation allowed for a match
C
      COMMON /LOCN/ LOCLAT, LOCLNG, RADIUS
C
      INTEGER IBASE(5), ILEVL(3), IMODE, ISTAT(10), LIST
      INTEGER ITDTE(3), ITTST(3), ITSAR(3), ITMCC(3), ITSPD(3)
      INTEGER IDLUT(3), IDTST(3), IDSAR(3), IDMCC(3)
C
C     these are the parameters required by the IMAGE subroutines
C         IT--- is the ITEM parameter for each of the files accessed
C         ID--- is the ID parameter for each of the files
C
      INTEGER SBUF(68), LBUF(98), MBUF(76), FBUF(50)
C
C         SBUF   - buffer for the SAR incident detail file
C         LBUF   - buffer for the LUT location detail file
C         MBUF   - buffer for the CMCC location detail file
C         FBUF   - buffer for the field test master file
C
      COMMON /BASE/ IBASE, ILEVL, IMODE, ISTAT, LIST, ITDTE,
     -              ITTST, ITSAR, ITMCC, ITSPD, IDLUT, IDMCC,
     -              IDSAR, IDTST, SBUF, LBUF, MBUF, FBUF
C
      COMMON/SELCT/ALLUT,ALSAT,SAT,SX,CX,SPESAT,PRIM,SECOND,THIRD,
     1     FOURTH,PST,SST,TST,FST,UNIT,DIF,BENT,REALT,COS4,CBC12,
     1     CBC24,ALLFR,POSN,START,EXIT,TAPE,SPECIF,SEVENT,PASS,
     1     IMAGE,CONTRY,IUSER,TYPE,ELTANG
      INTEGER CONTRY(3),IUSER,TYPE(4)
      REAL ELTANG
      LOGICAL ALLUT,ALSAT,SAT,SX,CX,PRIM,SECOND,THIRD,FOURTH
      LOGICAL IMAGE
      LOGICAL BENT,REALT,COS4,CBC12,CBC24,ALLFR,EXIT,TAPE,SPECIF
      INTEGER SPESAT(5),PST,SST,TST,FST,UNIT,POSN,START
      INTEGER PASS(4),SEVENT
      REAL DIF
C
C     ALLUT  - flag to indicate if all locations desired
C     ALSAT  - flag to indicate if all satellites desired
C     SAT    - flag to indicate that a specific satellite is desired
C     SX     - flag to indicate that all SARSAT satellites desired
C     CX     - flag to indicate that all COSPAS satellites desired
C     SPESAT - 5 element array containing the names of the specific
C              satellites desired
C     PRIM   - flag to indicate that the primary output was chosen
C     SECOND - flag to indicate that the second output was chosen
C     THIRD  - flag to indicate that the third output was chosen
C     FOURTH - flag to indicate that the fourth output was chosen
C     PST    - flag to indicate if the headers for the primary output
C              have been displayed yet
C     SST    - flag to indicate if the headers for the second output
C              have been displayed yet
C     TST    - flag to indicate if the headers for the third output
C              have been displayed yet
C     FST    - flag to indicate if the headers for the fourth output
C              have been displayed yet
C     UNIT   - variable containing the LU for the output
C     DIF    - variable containing the calculated difference between
C              the calculated location and the true location
C     BENT   - flag to indicate that 406 'bent pipe' data desired
C     REALT  - flag to indicate that 406 2.4 Kb/s real time data desired
C     COS4   - flag to indicate that 406 2.4 Kb/s COSPAS stored data desired
C     CBC12  - flag to indicate that CBC12i data desired
C     CBC24  - flag to indicate that CBF243 data desired
C     ALLFR  - flag to indicate that all the frequencys are desired
C     EXIT   - flag to indicate if the user wishes to abort the program
C     TAPE   - flag to indicate if output to go to magnetic tape
C     SPECIF - flag to indicate if a specific SATPAS and EVENT desired
C     SEVENT - event number of the specific SATPAS desired
C     PASS   - array containing the specific SATPAS desired
C
```

APPENDIX B

SOURCE CODE LISTINGS

THIRD OUTPUT
PRIMARY LOCATION

	DRIFT	CTAI	TCAI	MAJAX	MINAX	AMEAN	BIASI
1)	0.0000	20.0000	426.4746	84.062	36.784	0.0314	12500.
2)	0.0000	18.0000	548.4746	46.054	34.378	-0.5669	12500.
3)	4.0011	4.0000	574.4746	7.829	1.814	0.0021	12200.
4)	0.0000	6.0000	536.4746	13.052	6.166	-0.0085	12400.
5)	0.0000	3.0000	586.4746	25.975	9.877	0.0352	12250.
6)	0.0000	4.0000	578.4746	9.334	3.818	-0.2506	12000.
7)	0.0000	20.0000	318.4697	54.693	23.877	-0.4271	13350.
8)	1.5644	5.0000	506.4746	18.845	6.975	-0.0076	12350.
9)	0.5311	14.0000	522.4697	14.119	6.255	-0.0080	12200.
10)	0.0000	5.0000	536.4746	6.145	2.984	0.2373	12250.
11)	0.0000	20.0000	396.4746	24.806	15.868	-0.0110	10150.
12)	0.0000	20.0000	492.4746	267.980	203.003	-0.0740	20800.
13)	0.0000	20.0000	458.4746	33.923	18.861	0.0920	13650.
14)	7.3387	9.0000	442.4746	13.583	5.589	0.0142	12200.
15)	0.0000	3.0000	470.4746	17.060	6.816	-0.0024	9950.
16)	0.0188	6.0000	419.4746	56.680	15.020	0.0138	12600.
17)	0.0000	3.0000	409.4746	24.820	11.220	-3.0974	12800.
18)	0.0000	5.0000	423.4697	22.782	8.640	0.0240	10800.

FOURTH OUTPUT
PRIMARY LOCATION

	VARCTA	VARTCA	VARBIA	VARDI	CORCT	CORCB	CORCD	CORTB	CORTD	CORBD
1)	0.1533	4.3796	26.2943	0.0000	-0.6061	0.6179	0.0000	-0.9809	0.0000	0.0000
2)	0.1130	2.4750	15.3921	0.0000	-0.2465	0.4564	0.0000	-0.9445	0.0000	0.0000
3)	0.0235	0.0960	1.7623	0.1912	0.1341	0.6653	-0.7221	-0.1229	-0.2186	-0.8996
4)	0.0391	0.3250	3.7128	0.0000	0.0966	-0.0699	0.0000	-0.9046	0.0000	0.0000
5)	0.0776	0.5382	7.5539	0.0000	-0.2522	0.2824	0.0000	-0.9663	0.0000	0.0000
6)	0.0278	0.2185	1.8783	0.0000	0.2819	-0.3331	0.0000	-0.8457	0.0000	0.0000
7)	0.1007	2.6563	16.5740	0.0000	0.6089	-0.6570	0.0000	-0.9902	0.0000	0.0000
8)	0.0566	0.3661	9.0059	0.0738	-0.0489	0.5369	-0.6434	-0.5429	0.1166	-0.8754
9)	0.0422	0.3535	6.5291	0.7174	0.1672	-0.9267	0.8324	-0.2834	-0.1962	-0.8761
10)	0.0184	0.1556	1.5581	0.0000	-0.0886	-0.0098	0.0000	-0.8563	0.0000	0.0000
11)	0.0478	1.4305	7.5417	0.0000	-0.0754	0.1966	0.0000	-0.9573	0.0000	0.0000
12)	0.6158	15.5563	81.4650	0.0000	0.1052	0.0222	0.0000	-0.9498	0.0000	0.0000
13)	0.0618	1.8889	11.3546	0.0000	-0.3296	0.3722	0.0000	-0.9790	0.0000	0.0000
14)	0.0486	0.2812	6.2173	0.7687	0.0616	-0.6846	0.7538	-0.4023	-0.0204	-0.8720
15)	0.0509	0.3548	5.8171	0.0000	-0.2525	0.2916	0.0000	-0.8976	0.0000	0.0000
16)	0.1704	0.7565	19.8585	2.7615	0.0321	-0.5932	0.7662	-0.3588	0.0144	-0.8866
17)	0.0745	0.5669	7.2083	0.0000	0.1323	-0.4067	0.0000	-0.8597	0.0000	0.0000
18)	0.0689	0.6789	7.7781	0.0000	0.6928	-0.7947	0.0000	-0.9293	0.0000	0.0000

PRIMARY DATA

	DATE	SATPAS	MCCREF	EVENT	MESSNT	PRIMARY ELTLAT	PRIMARY ELTLONG	SECONDARY ELTLAT	SECONDARY ELTLONG	DIFF	DLAT	DLONG
LOCATION LATITUDE -		45.3500	LONGITUDE -		-75.8833	RADIUS -	400.0					
1)	840610	C1 09723	0	3	0 0	45.5585	-80.7893			383.7435	23.2102	-382.3840
2)	840610	C1 09723	0	4	1 0	42.1694	-77.6118			388.3337	-354.0545	-142.6086
3)	840610	C1 09724	0	1	1 0	42.0826	-74.8075			287.8038	-274.6637	87.7512
4)	840610	C1 09724	0	6	1 0	44.7706	-71.5335			348.0154	-64.4969	343.7547
5)	840610	C1 09724	0	8	1 0	42.4026	-76.3357			330.1027	-328.0950	-37.1062
6)	840610	C1 09724	0	12	0 0	42.8927	-74.3679			299.1360	-273.5392	123.5073
7)	840610	C1 09725	0	6	1 0	47.0348	-76.7996			200.3983	187.5474	-69.5192
8)	840610	C1 09731	0	2	1 0	44.9439	-71.6047			338.9100	-45.2054	337.1110
9)	840610	C1 09732	0	2	1 0	44.9463	-71.7553			327.1570	-44.9385	325.2322
10)	840610	C2 06082	0	8	1 0	45.0164	-71.7700			324.8330	-37.1346	323.6771
11)	840610	C2 06083	0	3	1 0	42.5985	-78.0126			350.5590	-306.2883	-174.4793
12)	840610	C2 06083	0	17	0 0	47.3617	-72.9559			317.3442	223.9376	220.7341
13)	840610	C2 06087	0	1	1 0	42.9296	-77.3849			294.9202	-269.4312	-122.3888
14)	840610	S1 06246	0	8	1 0	45.1289	-71.8908			313.8870	-24.6122	313.5546
15)	840610	S1 06246	0	12	1 0	45.9538	-80.2401			345.5715	67.2135	-337.1811
16)	840610	S1 06246	0	21	1 16	43.4048	-75.7534			216.7848	-216.5337	10.5052
17)	840610	S1 06246	0	23	1 16	42.4972	-78.8489			396.6629	-317.5646	-243.4024
18)	840610	S1 06253	0	17	1 16	47.2929	-75.5302			217.9788	216.2782	26.6599

SECOND OUTPUT

	CTA	PRIMARY LOCATION POINTS	SDEV	TREND	QUAL	PROB	NMWLS	TCA	QTIME	LOSTIM	BIAS	CORR	SCORE	
1)	-18.0315	52	36.4722	24.6196	172	49	4	5.2310	5.1081	5.3714	12464	8224	8224	18944	14
2)	-17.4622	110	37.7300	30.3439	454	50	2	5.2526	5.1081	5.3714	12625	8224	8224	20992	12
3)	3.6312	402	10.1961	0.4048	1777	53	2	7.0304	6.8714	7.1647	12139	8224	8224	18560	16
4)	6.8688	136	10.4365	15.2385	482	51	5	7.0217	6.8714	7.1647	12291	8224	8224	24576	14
5)	2.4750	98	19.0156	11.0717	265	51	2	7.0324	6.8714	7.1647	12312	8224	8224	30720	12
6)	3.9543	151	12.1843	6.6799	415	47	2	7.0304	6.8714	7.1647	11945	8224	8224	29184	12
7)	20.4584	147	24.4523	20.9708	684	52	4	8.7580	8.6614	8.8697	13088	8224	8224	23808	14
8)	4.9087	78	15.0764	6.2229	237	50	1	18.8742	18.7336	19.0178	12296	8224	8224	27648	12
9)	-13.3158	151	7.3081	2.7413	471	52	1	20.6490	20.5047	20.7867	12185	8224	8224	25856	14
10)	-4.3933	160	10.0374	3.9788	502	52	2	1.5300	1.3808	1.6756	12207	8224	8224	20992	14
11)	-18.9031	219	29.0866	22.6928	857	53	4	3.2906	3.1878	3.4256	10257	8224	8224	31488	14
12)	-19.6031	16	93.7124	35.0126	33	50	1	3.3225	3.1878	3.4256	20815	8224	8224	22016	12
13)	-19.1407	234	34.2184	31.4531	1004	48	4	18.6793	18.5467	18.7917	13474	8224	8224	24064	14
14)	-8.0771	123	17.3065	3.3026	341	51	1	0.6134	0.4911	0.7422	12135	8224	8224	25088	12
15)	-2.4411	77	19.1777	9.3834	253	49	-5	0.6219	0.4911	0.7422	9905	8224	8224	20480	12
16)	-5.0754	68	44.2487	21.4827	157	49	1	0.6074	0.4911	0.7422	12627	8224	8224	28672	12
17)	-2.6701	69	24.9144	13.2868	159	51	1	0.6053	0.4911	0.7422	12734	8224	8224	24064	12
18)	-4.5977	52	17.0331	3.6315	259	51	3	12.8471	12.7292	12.9822	10746	8224	8224	25088	12

APPENDIX A

SAMPLE RUN

```
:LOCAT
 Enter the name of the database to be used
SAR2:HW
Enter date for start and end of search: ie. YYMMDD YYMMDD
840610 840610
Specify retrieval output device LU: 10



       Specific SAT   $   A location        Cx              Sx
       Satpas         $ END of select                      EXIT

Enter target location data:
Specify the search location
Longitude? -75.8833
Latitude? 45.35
Radius?400


 ENTER THE REGION AND CASE NUMBER OF THIS LOCATION.
 THE CODES FOR THE REGIONS ARE:  HZ - HALIFAX,
TR -TRENTON, ED - EDMONTON, YJ -VICTORIA.
 ENTER THEM ACCORDING TO THIS EXAMPLE:  TR 1111



     $ Primary Output $  Second Output $ Third Output   $  Fourth Output
        Tape            END of select    Image Data         EXIT

Which output do you wish?



       406 Bent Pipe    406 Real Time     406 Cospas    $    CBC121
     $  CBC243          END of select     All frequencys      EXIT

Which output do you wish?



       406 Bent Pipe    406 Real Time     406 Cospas    $    CBC121
     $  CBC243        $ END of select     All frequencys      EXIT

Which output do you wish?

 There are   18 hits.
 Do you wish another query?
NO
```

5.0 SUMMARY COMMENTS

The computer program, LOCAT, has been described along with the USER'S GUIDE and instructions on how to compile and load the program. Further documentation is provided in the comments in the program listings, see Appendix B.

- The soft keys allowing the choice of output(s) are displayed.

- The soft keys allowing the choice of frequency are displayed. If the 406 MHz data is requested, a prompt for the country and specific beacon identification of the ELT is displayed.

- When the data retrieval is complete, the following message appears on the console to inform the user of the number of records retrieved:

```
There were XX hits.
Do you wish another query?.
```

  where XX is the number of records retrieved.

The option is provided to recycle and input another query. If another query is chosen, the program starts again. If another query is not desired, the retrieved data is written to the logical unit previously indicated for output and the program halts.

A sample run is given in Appendix A.

4.0 COMPILING AND LOADING

All the routines, relocatable modules and tranfer files related to LOCAT are found on cartridge 148 on the SEF disc. The source code is found in files starting with an & and the relocatable modules are found in files starting with a %. For example, the main program, LOCAT, has its source code in the file &LOCAT and its relocatable module in %LOCAT.

The main program and the subroutines are compiled using the FORTRAN 4X compiler. The transfer files ;LOCAT, and ;ALOCS have been created to load the main program and its related subroutines. To load the programs, enter:

```
:RU,LOADR,;LOCAT
:RU,LOADR,;ALOCS
```

In order to save the programs so that it does not have to be loaded each time the computer is logged on, enter:

```
:SP,LOCAT::113
:SP,ALOCS::113
```

VARCTA - standard deviation of CTA in degrees.
VARTCA - standard deviation of TCA in seconds.
VARBIA - standard deviation of BIAS in Hz.
VARDI - standard deviation of DRIFT in Hz/min.
CORCT - correlation of CTA with TCA.
CORCB - correlation of CTA with BIAS.
CORCD - correlation of CTA with DRIFT.
CORTB - correlation of TCA with BIAS.
CORTD - correlation of TCA with DRIFT.
CORBD - correlation of BIAS with DRIFT.

The end of the selection phase for each of these three data retrieval criteria, Satellite and Location, Frequency, and Output are indicated to the program by pressing the END OF SELECT soft key. Once all these criteria have been specified, the program then goes and searches the database for the records that meet these criteria.

3.0 USER'S GUIDE

In order to run this program, logon to the DEVELOP.SEF account on the SEF disk and enter :LOCAT. The following is the sequence of events that the program will go through.

- The name of the database being used is requested.
- A prompt is displayed requesting the start and end date that the data retrieval is to take place in. These dates are entered on the same line in the format:

  YYMMDD YYMMDD

- A prompt is displayed requesting identification of the logical unit to which the output is to be directed.
- The soft keys allowing the choice of location and satellite(s) are displayed.

```
FTN4
$FILES(15,15)
      SUBROUTINE OUTPT
     -          SEF-40-00  V00  (840726.0909)
      IMPLICIT NONE
C                                                                              *
C-------------------------------------------------------------------------------*
C                                                                              *
C   SARSAT EVALUATION FACILITY                                                 *
C                                                                              *
C-------------------------------------------------------------------------------*
C                                                                              *
C   DATE         VERSION       DESCRIPTION              AUTHOR                 *
C  83/10/05        00            -------             SUZANNE Y. SLINN         *
C                                                                              *
C-------------------------------------------------------------------------------*
C                                                                              *
C   DESCRIPTION:                                                               *
C     This subroutine determines which of the four output formats are          *
C   desired.                                                                   *
C                                                                              *
C   CALLING SEQUENCE:                                                          *
C                                                                              *
C     CALLED FROM: LOCAT                                                       *
C                                                                              *
C       CALL OUTPT                                                             *
C                                                                              *
C-------------------------------------------------------------------------------*
C
C
C-------------------------------------------------------------------------------*
C                                                                              *
C                         DATA DECLARATIONS                                    *
C                                                                              *
C-------------------------------------------------------------------------------*
C
C
C
      COMMON /LOCG/ LODAT(3), HIDAT(3), STDAT, ENDAT,OUTLU(5),RANGE
      INTEGER LODAT, HIDAT, OUTLU, RANGE
      DOUBLE PRECISION STDAT, ENDAT
C
      REAL LOCLAT, LOCLNG, RADIUS
C
C         LOCLAT - latitude of specified location
C         LOCLNG - longitude of specified location
C         RADIUS - maximum distance between specified location and
C                  file locations
C
      COMMON /LOCN/ LOCLAT, LOCLNG, RADIUS
C
      INTEGER IBASE(5), ILEVL(3), IMODE, ISTAT(10), LIST
      INTEGER ITDTE(3), ITTST(3), ITSAR(3), ITMCC(3), ITSPD(3)
      INTEGER IDLUT(3), IDTST(3), IDSAR(3), IDMCC(3)
C
      INTEGER SBUF(68), LBUF(98), MBUF(76), FBUF(50)
C
      COMMON /BASE/ IBASE, ILEVL, IMODE, ISTAT, LIST, ITDTE,
     -              ITTST, ITSAR, ITMCC, ITSPD, IDLUT, IDMCC,
     -              IDSAR, IDTST, SBUF, LBUF, MBUF, FBUF
C
      DOUBLE PRECISION QTIME, SDT
C
C         SDT    - start date
C
C
C         LODAT  - ascii start date YYMMDD
C         STDAT  - low date in seconds from 1980
C         ENDAT  - high date in seconds from 1980
C
C         OUTLU  - output device, LU or filename
C
      INTEGER RBUF(40), IPARM(5), PBUF(33)
      INTEGER LU, RET
      INTEGER SYS, ISYS, LOGLU
C
C         RBUF   - receiving buffer
C         IPARM  - return buffer for RMPAR call
C         PBUF   - return buffer from PARSE
C
      REAL ONEDAY
      INTEGER DATCH(3), LOOP, DUM(7), I1
      INTEGER UNIT
C
C
      LOGICAL LUT, MCC, SAR, TEST, ALL
C
C         LUT    - flag to indicate LUT location detail file selected
C         MCC    - flag to indicate MCC location detail file selected
C         SAR    - flag to indicate SARSTAT incident detail file selected
C         TEST   - flag to indicate FIELD test master file selectedlected
C         ALL    - all files will be searched
C
```

```
      INTEGER KEYS(8,8), PROM(8), RK
      INTEGER FLG, AA, IFBRK
C
C       FLG   - flag to indicate if /E used
C       AA    - dummy parameter for IFBRK
C
C
      COMMON/SELCT/ALLUT,ALSAT,SAT,SX,CX,SPESAT,PRIM,SECOND,THIRD,
     1      FOURTH,PST,SST,TST,FST,UNIT,DIF,BENT,REALT,COS4,CBC12,
     1      CRC24,ALLFR,POSN,START,EXIT,TAPE,SPECIF,SEVENT,PASS,IMAGE,
     1      CONTRY,IUSER,TYPE,ELTANG
      INTEGER CONTRY(3),IUSER,TYPE(4)
      REAL ELTANG
      LOGICAL ALLUT,ALSAT,SAT,SX,CX,PRIM,SECOND,THIRD,FOURTH
      LOGICAL BENT,REALT,COS4,CBC12,CBC24,ALLFR,EXIT,TAPE,SPECIF
      LOGICAL IMAGE
      INTEGER SPESAT(5),I,IOS,PST,SST,TST,FST,UNIT,POSN,START
      INTEGER SEVENT,PASS(4)
      REAL DIF
C
C       ALLUT -flag to indicate if all locations wanted
C       ALSAT -flag to indicate if all satellites wanted
C       SAT   -flag to indicate if specific satellites wanted
C       SX    -flag to indicate if all SARSAT satellites wanted
C       CX    -flag to indicate if all COSPAS satellites wanted
C       SPESAT-array that contains the specific satellites desired
C
      DATA KEYS /'  Primary Output   Second Output',
     -           '  Third Output    Fourth Output',
     -           '    Tape          END of select ',
     -           '  Image Data          EXIT      '/
C
      DATA PROM /'f1f2f3f4f5f6f7f8'/
C       KEYS    - array containing contents of soft keys
C       PROM    - array of values to return for soft keys
C       RK      - actual value returned
C
C*************************************************************************
C
C                EXECUTABLE  CODE
C
C*************************************************************************
C
      ISYS = LOGLU(SYS)
C
C       initialize variables
C
C
      LU = 1
      UNIT = 99
C
C       search files specified
C
  100 CONTINUE
      CALL FUNKY (KEYS,PROM)
      WRITE (LU,1003)
  101 CALL REIO(1,LU,RK,-2)
      RK = RK -2Hf0
      IF (RK .LT. 1 .OR. RK .GT. 8) GO TO 101
      IF (RK .EQ. 8) THEN
         EXIT=.TRUE.
         GO TO 999
      ENDIF
C
C      set * in key selected and check for further selects
C
      IF (KEYS(1,RK) .EQ. 2H  ) THEN
          KEYS(1,RK) = 2H*
          IF (RK .EQ. 1)     PRIM= .TRUE.
          IF (RK .EQ. 2)     SECOND =.TRUE.
          IF (RK .EQ. 3)  THIRD=.TRUE.
          IF (RK .EQ. 4) FOURTH=.TRUE.
          IF (RK .EQ. 5) TAPE = .TRUE.
          IF (RK .EQ. 7) IMAGE = .TRUE.
          GO TO 102
        ENDIF
C
      IF (KEYS(1,RK) .EQ. 2H* ) THEN
          KEYS(1,RK) = 2H
          IF (RK .EQ. 1)     PRIM= .FALSE.
          IF (RK .EQ. 2)     SECOND=.FALSE.
          IF (RK .EQ. 3)  THIRD=.FALSE.
          IF (RK .EQ. 4) FOURTH=.FALSE.
          IF (RK .EQ. 5) TAPE = .FALSE.
          IF (RK .EQ. 7) IMAGE = .FALSE.
        ENDIF
C
  102 CALL FUNKY(KEYS(1,RK),RK)
      IF (RK .EQ. 6)  GO TO 200
      GO TO 101
C
  200 CONTINUE
```

```
C
C     change '$' to ' '
C
      DO 4 RK=1,8
       KEYS(1,RK)=2H
 4    CONTINUE
C
C     open the header file
C
      OPEN(14,FILE='HEADR',IOSTAT=IOS,STATUS='OLD')
      IF (IOS .NE. 0) WRITE(LU,1006)IOS
      IF (IOS .NE. 0) GO TO 999
C
 999  CONTINUE
      WRITE(LU,1005)
 1000 FORMAT(' ERROR IN OPENING PRIMARY. IOS IS ',I4)
 1001 FORMAT(' ERROR IN OPENING SECOND. IOS IS ',I4)
 1002 FORMAT(' ERROR IN OPENING THIRD.  IOS IS ',I4)
 1003 FORMAT(' Which output do you wish?')
 1004 FORMAT(' ERROR IN OPENING FOURTH. IOS IS ',I4)
 1005 FORMAT(' ')
 1006 FORMAT(' ERROR IN OPENING HEADR. IOS IS ',I4)
      RETURN
      END
```

```
FTN4
$FILES(15,15)
      SUBROUTINE OUTP2(PLACE)
     -          SEF-40-00  V00  (840725.1426)
      IMPLICIT NONE
C*****************************************************************
C*                                                               *
C*  SARSAT EVALUATION FACILITY                                   *
C*                                                               *
C*---------------------------------------------------------------*
C*                                                               *
C*  DATE        VERSION      DESCRIPTION          AUTHOR         *
C*                                                               *
C*  83/10/05      00           -------          SUZANNE Y. SLINN *
C*                                                               *
C*---------------------------------------------------------------*
C*                                                               *
C*  DESCRIPTION:                                                 *
C*     This subroutine opens the output files.                   *
C*                                                               *
C*  CALLING SEQUENCE:                                            *
C*                                                               *
C*    CALLED FROM: LOCAT                                         *
C*                                                               *
C*      CALL OUTPT                                               *
C*                                                               *
C*---------------------------------------------------------------*
C*                                                               *
C*                                                               *
C*---------------------------------------------------------------*
C*                                                               *
C*                      DATA DECLARATIONS                        *
C*                                                               *
C*---------------------------------------------------------------*
C*                                                               *
C*                                                               *
C
      COMMON /LOGG/ LODAT(3), HIDAT(3), STDAT, ENDAT,OUTLU(5),RANGE
      INTEGER PLACE
      INTEGER LODAT, HIDAT, OUTLU, RANGE
      DOUBLE PRECISION STDAT, ENDAT
C
      REAL LOCLAT, LOCLNG, RADIUS
C
C*        LOCLAT - latitude of specified location
C*        LOCLNG - longitude of specified location
C*        RADIUS - maximum distance between specified location and
C*                 file locations
C
      COMMON /LOCN/ LOCLAT, LOCLNG, RADIUS
C
      INTEGER IBASE(5), ILEVL(3), IMODE, ISTAT(10), LIST
      INTEGER ITDTE(3), ITTST(3), ITSAR(3), ITMCC(3), ITSPD(3)
      INTEGER IDLUT(3), IDTST(3), IDSAR(3), IDMCC(3)
C
```

```
      INTEGER SBUF(68), LBUF(98), MBUF(76), FBUF(50)
C
      COMMON /BASE/ IBASE, ILEVL, IMODE, ISTAT, LIST, ITDTE,
     -              ITTST, ITSAR, ITMCC, ITSPD, IDLUT, IDMCC,
     -              IDSAR, IDTST, SBUF, LBUF, MBUF, FBUF
C
      DOUBLE PRECISION QTIME, SDT
C
C         SDT    - start date
C
C
C         LODAT  - ascii start date YYMMDD
C         STDAT  - low date in seconds from 1980
C         ENDAT  - high date in seconds from 1980
C
C         OUTLU  - output device, LU or filename
C
      INTEGER RBUF(40), IPARM(5), PBUF(33)
      INTEGER LU, RET
      INTEGER SYS, ISYS, LOGLU
C
C         RBUF   - receiving buffer
C         IPARM  - return buffer for RMPAR call
C         PBUF   - return buffer from PARSE
C
      REAL ONEDAY
      INTEGER DATCH(3), LOOP, DUM(7), I1
      INTEGER UNIT
C
C
      LOGICAL LUT, MCC, SAR, TEST, ALL
C
C         LUT    - flag to indicate LUT location detail file selected
C         MCC    - flag to indicate MCC location detail file selected
C         SAR    - flag to indicate SARSTAT incident detail file selected
C         TEST   - flag to indicate FIELD test master file selectedlected
C         ALL    - all files will be searched
C
      INTEGER KEYS(8,8), PROM(8), RK
      INTEGER FLG, AA, IFBRK
C
C       FLG    - flag to indicate if /E used
C       AA     - dummy parameter for IFBRK
C
C
      COMMON/SELCT/ALLUT,ALSAT,SAT,SX,CX,SPESAT,PRIM,SECOND,THIRD,
     1      FOURTH,PST,SST,TST,FST,UNIT,DIF,BENT,REALT,COS4,CBC12,
     1      CBC24,ALLFR,POSN,START,EXIT,TAPE,SPECIF,SEVENT,PASS,IMAGE,
     1      CONTRY,IUSER,TYPE,ELTANG
      INTEGER CONTRY(3),IUSER,TYPE(4)
      REAL ELTANG
      LOGICAL ALLUT,ALSAT,SAT,SX,CX,PRIM,SECOND,THIRD,FOURTH
      LOGICAL BENT,REALT,COS4,CBC12,CBC24,ALLFR,EXIT,TAPE,SPECIF
      LOGICAL IMAGE
      INTEGER SPESAT(5),I,IOS,PST,SST,TST,FST,UNIT,POSN,START
      INTEGER SEVENT,PASS(4)
      REAL DIF
C
C       ALLUT -flag to indicate if all locations wanted
C       ALSAT -flag to indicate if all satellites wanted
C       SAT   -flag to indicate if specific satellites wanted
C       SX    -flag to indicate if all SARSAT satellites wanted
C       CX    -flag to indicate if all COSPAS satellites wanted
C       SPESAT-array that contains the specific satellites desired
C
C*************************************************************************
C
C                EXECUTABLE  CODE
C
C*************************************************************************
C
      ISYS = LOGLU(SYS)
C
C         initialize variables
C
C
      LU = 1
      UNIT = 99
C
C       search files specified
C
  100 CONTINUE
  200 CONTINUE
C
      OPEN(14,FILE='HEADR',IOSTAT=IOS,STATUS='OLD')
      IF (IOS .NE. 0) WRITE(LU,1006)IOS
      IF (IOS .NE. 0) GO TO 999
```

```
C
C
C     open primary file if this output is desired
C
      IF (PRIM) THEN
        OPEN(10,FILE='PRIMARY',IOSTAT=IOS,STATUS='OLD')
        IF (IOS .NE. 0) WRITE(LU,1000)IOS
        IF (IOS .NE. 0) GO TO 999
      ENDIF
C
C     open second file if this output is desired
C
      IF (SECOND) THEN
        OPEN(11,FILE='SECOND',IOSTAT=IOS,STATUS='OLD')
        IF (IOS .NE. 0) WRITE(LU,1001)IOS
        IF (IOS .NE. 0) GO TO 999
      ENDIF
C
C     open third file if this output is desired
C
      IF (THIRD) THEN
        OPEN(12,FILE='THIRD',IOSTAT=IOS,STATUS='OLD')
        IF (IOS .NE. 0) WRITE(LU,1002)IOS
        IF (IOS .NE. 0) GO TO 999
      ENDIF
C
C     open fourth file if this output is desired
C
      IF (FOURTH) THEN
        OPEN(13,FILE='FOURTH',IOSTAT=IOS,STATUS='OLD')
        IF (IOS .NE. 0) WRITE(LU,1004)IOS
        IF (IOS .NE. 0)GO TO 999
      ENDIF
      IF (IMAGE ) THEN
        IF (PRIM) THEN
          OPEN(21,FILE='IPRIMARY',IOSTAT=IOS,STATUS='OLD')
          IF (IOS .NE. 0) WRITE(LU,1000)IOS
          IF (IOS .NE. 0) GO TO 999
        ENDIF
        IF (SECOND) THEN
            OPEN(22,FILE='ISECOND',IOSTAT=IOS,STATUS='OLD')
            IF (IOS .NE. 0) WRITE(LU,1001)IOS
            IF (IOS .NE. 0) GO TO 999
        ENDIF
        IF (THIRD) THEN
            OPEN(23,FILE='ITHIRD',IOSTAT=IOS,STATUS='OLD')
            IF (IOS .NE. 0) WRITE(LU,1001)IOS
            IF (IOS .NE. 0) GO TO 999
        ENDIF
        IF (FOURTH) THEN
            OPEN(24,FILE='IFOURTH',IOSTAT=IOS,STATUS='OLD')
            IF (IOS .NE. 0) WRITE(LU,1001)IOS
            IF (IOS .NE. 0) GO TO 999
        ENDIF
      ENDIF
C
 999  CONTINUE
      IF (POSN .NE. 1)THEN
        DO 65 I1=1,(POSN-1)
          IF (PRIM) READ(10,2017)I
          IF(SECOND) READ(11,2017)I
          IF (THIRD) READ(12,2017)I
          IF (FOURTH) READ(13,2017)I
          IF (IMAGE) THEN
            IF (PRIM) READ(21,2017)I
            IF (SECOND) READ(22,2017)I
            IF (THIRD) READ(23,2017)I
            IF (FOURTH) READ(24,2017)I
          ENDIF
 2017    FORMAT(A2)
 65      CONTINUE
       DO 96 I1=1,(PLACE-1)
         READ(14,2017)I
 96    CONTINUE
       ENDIF
C
      WRITE(LU,1005)
 1000 FORMAT(' ERROR IN OPENING PRIMARY. IOS IS ',I4)
 1001 FORMAT(' ERROR IN OPENING SECOND. IOS IS ',I4)
 1002 FORMAT(' ERROR IN OPENING THIRD.  IOS IS ',I4)
 1003 FORMAT(' Which output do you wish?')
 1004 FORMAT(' ERROR IN OPENING FOURTH. IOS IS ',I4)
 1005 FORMAT(' ')
 1006 FORMAT(' ERROR IN OPENING HEADR. IOS IS ',I4)
      RETURN
      END
```

```
FTN4
%FILES(15,15)
      SUBROUTINE FREQ
     -          SEF-40-00  V00  (840726.0911)
      IMPLICIT NONE
C
C---------------------------------------------------------------------*
C                                                                     *
C  SARSAT EVALUATION FACILITY                                         *
C                                                                     *
C---------------------------------------------------------------------*
C                                                                     *
C  DATE       VERSION          DESCRIPTION          AUTHOR           *
C 03/10/05      00               -------           SUZANNE Y. SLINN  *
C                                                                     *
C---------------------------------------------------------------------*
C                                                                     *
C DESCRIPTION:                                                        *
C    This subroutine determines which of the frequencies the user     *
C desires to use.                                                     *
C                                                                     *
C CALLING SEQUENCE:                                                   *
C                                                                     *
C    CALLED FROM: LOCAT                                               *
C                                                                     *
C       CALL FREQ                                                     *
C                                                                     *
C---------------------------------------------------------------------*
C
C
C---------------------------------------------------------------------*
C                                                                     *
C                        DATA DECLARATIONS                            *
C                                                                     *
C---------------------------------------------------------------------*
C
C
      COMMON /LOGG/ LODAT(3), HIDAT(3), STDAT, ENDAT,OUTLU(5),RANGE
      INTEGER LODAT, HIDAT, OUTLU, RANGE
      DOUBLE PRECISION STDAT, ENDAT
C
      REAL LOCLAT, LOCLNG, RADIUS
C
C         LOCLAT - latitude of specified location
C         LOCLNG - longitude of specified location
C         RADIUS - maximum distance between specified location and
C                  file locations
C
      COMMON /LOCN/ LOCLAT, LOCLNG, RADIUS
C
      INTEGER IBASE(5), ILEVL(3), IMODE, ISTAT(10), LIST
      INTEGER ITDTE(3), ITTST(3), ITSAR(3), ITMCC(3), ITSPD(3)
      INTEGER IDLUT(3), IDTST(3), IDSAR(3), IDMCC(3)
C
      INTEGER SBUF(68), LBUF(98), MBUF(76), FBUF(50)
C
      COMMON /BASE/ IBASE, ILEVL, IMODE, ISTAT, LIST, ITDTE,
     -              ITTST, ITSAR, ITMCC, ITSPD, IDLUT, IDMCC,
     -              IDSAR, IDTST, SBUF, LBUF, MBUF, FBUF
C
      DOUBLE PRECISION QTIME, SDT
C
C         SDT    - start date
C
C
C         LODAT  - ascii start date YYMMDD
C         STDAT  - low date in seconds from 1980
C         ENDAT  - high date in seconds from 1980
C
C         OUTLU  - output device, LU or filename
C
      INTEGER RBUF(40), IPARM(5), PBUF(33)
      INTEGER LU, RET
      INTEGER SYS, ISYS, LOGLU
C
C         RBUF   - receiving buffer
C         IPARM  - return buffer for RMPAR call
C         PBUF   - return buffer from PARSE
C
```

```
      REAL ONEDAY
      INTEGER DATCH(3), LOOP, DUM(7), I1
      INTEGER UNIT
C
C
      LOGICAL LUT, MCC, SAR, TEST, ALL
C
C         LUT    - flag to indicate LUT location detail file selected
C         MCC    - flag to indicate MCC location detail file selected
C         SAR    - flag to indicate SARSTAT incident detail file selected
C         TEST   - flag to indicate FIELD test master file selectedlected
C         ALL    - all files will be searched
C
      INTEGER KEYS(8,6), PROM(8), RK
      INTEGER FLG, AA, IFBRK
C
C       FLG   - flag to indicate if /E used
C       AA    - dummy parameter for IFBRK
C
C
      COMMON/SELCT/ALLUT,ALSAT,SAT,SX,CX,SPESAT,PRIM,SECOND,THIRD,
     1      FOURTH,PST,SST,TST,FST,UNIT,DIF,BENT,REALT,COS4,
     1      CBC12,CBC24,ALLFR,POSN,START,EXIT,TAPE,SPECIF,SEVENT,PASS,
     1      IMAGE,CONTRY,IUSER,TYPE,ELTANG
      INTEGER CONTRY(3),IUSER,TYPE(4)
      REAL ELTANG
      LOGICAL IMAGE
      LOGICAL ALLUT,ALSAT,SAT,SX,CX,PRIM,SECOND,THIRD,FOURTH
      LOGICAL BENT,REALT,COS4,CBC12,CBC24,ALLFR,EXIT,TAPE,SPECIF
      INTEGER SPESAT(5),I,IOS,PST,SST,TST,FST,UNIT,POSN,START
      INTEGER SEVENT,PASS(4)
      REAL DIF
C
C       ALLUT -flag to indicate if all locations wanted
C       ALSAT -flag to indicate if all satellites wanted
C       SAT   -flag to indicate if specific satellites wanted
C       SX    -flag to indicate if all SARSAT satellites wanted
C       CX    -flag to indicate if all COSPAS satellites wanted
C       SPESAT-array that contains the specific satellites desired
C
      DATA KEYS /'  406 Bent Pipe   406 Real Time ',
     -           '   406 Cospas        CBC121    ',
     -           '   CBC243         END of select ',
     -           ' All frequencys      EXIT      '/
C
      DATA PROM /'f1f2f3f4f5f6f7f8'/
C        KEYS    - array containing contents of soft keys
C        PROM    - array of values to return for soft keys
C        RK      - actual value returned
C
C*****************************************************************************
C
C                 EXECUTABLE  CODE
C
C*****************************************************************************
C
      ISYS = LOGLU(SYS)
C
C         initialize variables
C
      ALLFR = .FALSE.
      BENT = .FALSE.
      REALT = .FALSE.
      COS4 = .FALSE.
      CBC12 = .FALSE.
      CBC24 = .FALSE.
      LU = 1
      UNIT = 99
C
C       search files specified
C
  100 CONTINUE
      CALL FUNKY (KEYS,PROM)
      WRITE (LU,1003)
  101 CALL REID(1,LU,RK,-2)
      RK = RK -2H30
      IF (RK .LT. 1 .OR. RK .GT. 8) GO TO 101
      IF (RK .EQ. 8)  THEN
        EXIT= TRUE.
        GO TO 999
      ENDIF
C
C       set 8 in key selected and check for further selects
C
      IF (KEYS(1,RK) .EQ. 2H  ) THEN
          KEYS(1,RK) = 2H8
          IF (RK .EQ. 1)     BENT= .TRUE.
          IF (RK .EQ. 2)     REALT = TRUE.
          IF (RK .EQ. 3)  COS4= TRUE.
          IF (RK .EQ. 4) CBC12= TRUE.
          IF (RK .EQ. 5) CBC24= TRUE.
          IF (RK .EQ. 7) ALLFR=.TRUE.
          GO TO 102
        ENDIF
```

```
C
      IF (KEYS(1,RK) .EQ. 2H$ ) THEN
          KEYS(1,RK) = 2H
          IF (RK .EQ. 1)     BENT= .FALSE.
          IF (RK .EQ. 2)     REALT=.FALSE.
          IF (RK .EQ. 3)   COS4=.FALSE.
          IF (RK .EQ. 4) CBC12=.FALSE.
          IF (RK .EQ. 5) CBC24=.FALSE.
          IF (RK .EQ. 7) ALLFR=.TRUE.
        ENDIF
C
  102 CALL FUNKY(KEYS(1,RK),RK)
      IF (RK .EQ. 6)  GO TO 200
      GO TO 101
C
  200 CONTINUE
C
C     clear the '$' from the select array
C
      DO 4 RK=1,8
       KEYS(1,RK)=2H
    4 CONTINUE
C
  999 CONTINUE
      WRITE(LU,1005)
 1000 FORMAT(' ERROR IN OPENING PRIMARY. IOS IS ',I4)
 1001 FORMAT(' ERROR IN OPENING SECOND. IOS IS ',I4)
 1002 FORMAT(' ERROR IN OPENING THIRD.  IOS IS ',I4)
 1003 FORMAT(' Which output do you wish?')
 1004 FORMAT(' ERROR IN OPENING FOURTH. IOS IS ',I4)
 1005 FORMAT(' ')
      RETURN
      END
```

```
FTN4
$FILES(15,15)
      SUBROUTINE OUTDT(ICONTR)
     -          ' SEF-40-00  V00  (840727.1242)
      IMPLICIT NONE
C
C*---------------------------------------------------------------------*
C*                                                                     *
C*   SARSAT EVALUATION FACILITY                                        *
C*                                                                     *
C*---------------------------------------------------------------------*
C*                                                                     *
C* DATE       VERSION          DESCRIPTION          AUTHOR             *
C*                                                                     *
C* 83/10/05     00               --------           SUZANNE Y. SLINN   *
C*                                                                     *
C*                                                                     *
C*---------------------------------------------------------------------*
C*                                                                     *
C* DESCRIPTION:                                                        *
C*          This subroutine formats the data from the four output      *
C*    files into the form required for output to the printer.  This    *
C*    routine calculates the delta lat and delta long if necessary.    *
C*                                                                     *
C* CALLING SEQUENCE:                                                   *
C*                                                                     *
C*  CALLED FROM: LOCAT                                                 *
C*    CALL OUTDT                                                       *
C*                                                                     *
C*---------------------------------------------------------------------*
C*                                                                     *
C*---------------------------------------------------------------------*
C*                                                                     *
C*                        DATA DECLARATIONS                            *
C*                                                                     *
C*---------------------------------------------------------------------*
C*
C*
C
      COMMON /LOGG/ LODAT(3), HIDAT(3), STDAT, ENDAT,OUTLU(5),RANGE
      INTEGER LODAT, HIDAT, OUTLU, RANGE
      DOUBLE PRECISION STDAT, ENDAT
C
      REAL LOCLAT, LOCLNG, RADIUS,LUTLA,LUTLG,DSTNC
C
C         LOCLAT - latitude of specified location
C         LOCLNG - longitude of specified location
C         RADIUS - maximum distance between specified location and
C                  file locations
C
```

```
      COMMON /LOCN/ LOCLAT, LOCLNG, RADIUS
C
      INTEGER IBASE(5), ILEVL(3), IMODE, ISTAT(10), LIST
      INTEGER ITDTE(3), ITTST(3), ITSAR(3), ITMCC(3), ITSPD(3)
      INTEGER IDLUT(3), IDTST(3), IDSAR(3), IDMCC(3)
C
      INTEGER SBUF(68), LBUF(98), MBUF(76), FBUF(50) ,BGBUF(200)
C
      COMMON /BASE/ IBASE, ILEVL, IMODE, ISTAT, LIST, ITDTE,
     -              ITTST, ITSAR, ITMCC, ITSPD, IDLUT, IDMCC,
     -              IDSAR, IDTST, SBUF, LBUF, MBUF, FBUF
C
      DOUBLE PRECISION QTIME, SDT
C
C         SDT    - start date
C
C
C         LODAT  - ascii start date YYMMDD
C         STDAT  - low date in seconds from 1980
C         ENDAT  - high date in seconds from 1980
C
C         OUTLU  - output device, LU or filename
C
      INTEGER RBUF(40), IPARM(5), PBUF(33)
      INTEGER LU, RET
      INTEGER SYS, ISYS, LOGLU
C
C         RBUF   - receiving buffer
C         IPARM  - return buffer for RMPAR call
C         PBUF   - return buffer from PARSE
C
      REAL ONEDAY
      INTEGER DATCH(3), LOOP, DUM(7), I1
      INTEGER UNIT
C
C
      LOGICAL LUT, MCC, SAR, TEST, ALL
C
C         LUT    - flag to indicate LUT location detail file selected
C         MCC    - flag to indicate MCC location detail file selected
C         SAR    - flag to indicate SARSTAT incident detail file selected
C         TEST   - flag to indicate FIELD test master file selectedlected
C         ALL    - all files will be searched
C
      INTEGER KEYS(8,8), PROM(8), RK
      INTEGER FLG, AA, IFBRK
C
C       FLG    - flag to indicate if /E used
C       AA     - dummy parameter for IFBRK
C
C
      COMMON/SELCT/ALLUT,ALSAT,SAT,SX,CX,SPESAT,PRIM,SECOND,THIRD,
     1       FOURTH,PST,SST,TST,FST,UNIT,DIF,BENT,REALT,COS4,CBC12,
     1       CBC24,ALLFR,POSN,START,EXIT,TAPE,SPECIF,SEVENT,PASS,
     1       IMAGE,CONTRY,IUSER,TYPE,ELTANG
      INTEGER CONTRY(3),IUSER,TYPE(4)
      REAL ELTANG
      LOGICAL ALLUT,ALSAT,SAT,SX,CX,PRIM,SECOND,THIRD,FOURTH
      LOGICAL BENT,REALT,COS4,CBC12,CBC24,ALLFR,EXIT,TAPE,SPECIF
      LOGICAL IMAGE
      INTEGER SPESAT(5),I,PST,SST,TST,FST,MSG1(120),POSN,START
      INTEGER SEVENT,PASS(4)
      REAL DIF
C
C       ALLUT -flag to indicate if all locations wanted
C       ALSAT -flag to indicate if all satellites wanted
C
C       SAT   -flag to indicate if specific satellites wanted
C       SX    -flag to indicate if all SARSAT satellites wanted
C       CX    -flag to indicate if all COSPAS satellites wanted
C       SPESAT-array that contains the specific satellites desired
C
      DOUBLE PRECISION LATC,LATT,LONGT,LONGC,FA,DIFFR,DLAT,DLONG
      REAL R,P
C
C       LATC  -calculated latitude from LUT file
C       LATT  -true latitude as input by the user
C       LONGT -true longitude as input by the user
C       LONGC -calculated longitude from the LUT file
C       DLAT  -difference in latitude in km.
C       DLONG -difference in longitude in km.
C
      INTEGER TRYA(6),TRYB(6),LINE,PAGE,RSPCE,NLINE,POS,CPOS
      INTEGER MPASS(4),MEVET,LPASS(4),LEVET,ITEM(3),MREF,MSG2(80)
      REAL LONGA,LONGB,LATA,LATB,FLAT,FLONG,FRADI,CRADI,CLAT,CLONG
      INTEGER ELTI1,ELTI2,ELTI3,ELTI4,SPOT,ICONTR(10)
        EQUIVALENCE(TRYA(1),MSG1(22))
        EQUIVALENCE(TRYB(1),MSG1(27))
C         KEYS    - array containing contents of soft keys
C         PROM    - array of values to return for soft keys
C         RK      - actual value returned
        DATA ITEM/6HMCCREF/
```

```
C
C*************************************************************************
C
C                EXECUTABLE  CODE
C
C*************************************************************************
C
      SPOT = 0
      CALL LGBUF(BGBUF,200)
      ISYS = LOGLU(SYS)
C
C        initialize variables
C
C
C     read from the header file
C
      REWIND (14)
      READ(14,FMT=1022,IOSTAT=ISTAT)POS,FLAT,FLONG,FRADI
      CPOS=POS
      CRADI=FRADI
      CLAT=FLAT
      CLONG=FLONG
  300 READ(14,FMT=1022,IOSTAT=ISTAT,END=121)POS,FLAT,FLONG,FRADI
      IF (CPOS .EQ. POS) THEN
         CPOS=POS
         CRADI=FRADI
         CLAT=FLAT
         CLONG=FLONG
         GO TO 300
      ENDIF
C
      IF (ISTAT .NE. 0) THEN
         WRITE(1,120)ISTAT
  120    FORMAT(' ERROR IN READING HEADER FILE, IOS IS ',I4)
         GO TO 999
      ENDIF
      GO TO 122
  121 POS=999
  122 CONTINUE
C
C
      PAGE=43
      IF (CRADI .EQ. 0.0) THEN
        LINE=7
      ELSE
        LINE=7
      ENDIF
      NLINE=1
C
      IF (PRIM) THEN
        P=3.141592654/180.
        R=6378.
        IF (PST .EQ. 1) GO TO 100
        PST=1
        WRITE(UNIT,1000)
        WRITE(UNIT,1001)
        WRITE(UNIT,1002)
        WRITE(UNIT,1)
        WRITE(UNIT,1023)CLAT,CLONG,CRADI
        WRITE(UNIT,1)
  100   CONTINUE
        REWIND(10,IOSTAT=ISTAT,ERR=993)
C
C     initialize input buffer with blanks
C
        I1=1
        CALL FILBS(400,240,MSG1,I1)
C
C     read in record from primary file
C
  101   CONTINUE
        READ(10,FMT=1004,IOSTAT=ISTAT,ERR=993,END=103)(MSG1(I1),I1=1,50)
C
        IF (ISTAT .NE. 0) THEN
          WRITE(1,102)ISTAT
  102     FORMAT(' READ ERROR, ISTAT IS ',I4)
          GO TO 999
  993     CONTINUE
          WRITE(1,992)ISTAT
  992     FORMAT(' ERROR IN READING FILE ',I4)
        ENDIF
        DECODE(9,1006,TRYA)LUTLA
        DECODE(9,1006,TRYB)LUTLG
C
C       calculate delta lat, delta long
C
        LATC=LUTLA*P
        LATT=CLAT*P
        LONGT=CLONG*P
        LONGC=LUTLG*P
C       if a specific location was specified then
        IF (CRADI .NE. 0.0) THEN
           DIFFR=DSTNC(CLONG,CLAT,LUTLG,LUTLA)
           DLAT=R*(LATC-LATT)
           DLONG=R*(LONGC-LONGT)*COS(LATC)
```

```
C       no specific location was specified
        ELSE
           DIFFR=0
           DLAT=0
           DLONG=0
        ENDIF
        WRITE(UNIT,1019)NLINE,(MSG1(I1),I1=1,48),DIFFR,DLAT,DLONG
        NLINE=NLINE+1
        LINE=LINE+1
 555    CONTINUE
C
C       if end of page, skip to next page
C
        IF (LINE .GE. PAGE) THEN
          RSPGE=PAGE-LINE
          DO 2 I1=1,RSPGE+8
            WRITE(UNIT,1)
 1          FORMAT(' ')
 2        CONTINUE
          WRITE(UNIT,1000)
          WRITE(UNIT,1001)
          WRITE(UNIT,1002)
          LINE=3
C         NLINE=1
        ENDIF
 225    CONTINUE
C
C       if have read all the records for this query then display
C       the next query location
C
        IF (NLINE .EQ. POS) THEN
            WRITE(UNIT,1)
            WRITE(UNIT,1023)FLAT,FLONG,FRADI
            WRITE(UNIT,1)
            LINE=LINE+3
            CRADI=FRADI
            CLAT=FLAT
            CLONG=FLONG
            READ(14,FMT=1022,IOSTAT=ISTAT,END=123)POS,FLAT,FLONG,FRADI
            IF (ISTAT .NE. 0) THEN
             WRITE(1,120)ISTAT
             GO TO 999
            ENDIF
            IF (LINE .GE. PAGE) GO TO 555
            GO TO 225
        ENDIF
C
        GO TO 101
 123  POS=999
      GO TO 101
      ENDIF
 103  CONTINUE
      IF (SECOND) THEN
        RSPGE=PAGE-LINE
        DO 4 I1=1,RSPGE+8
          WRITE(UNIT,1)
 4      CONTINUE
        LINE=3
        NLINE=1
      ENDIF
C
C     if second output is desired then display its headers
C
      IF (SECOND) THEN
       IF (SST .EQ. 1) GO TO 104
        SST=1
        WRITE(UNIT,1007)
        WRITE(UNIT,1008)
        IF ((BENT) .OR. (REALT) .OR. (COS4)) THEN
         WRITE(UNIT,2022)
        ELSE
         WRITE(UNIT,1009)
        ENDIF
 104    CONTINUE
        REWIND(11)
        REWIND(14)
        READ(14,FMT=1022)POS,FLAT,FLONG,FRADI
        CPOS = POS
        READ(14,FMT=1022)POS,FLAT,FLONG,FRADI
C
C       initialize buffer with blanks
C
        CALL FILBS(400,240,MSG1,I1)
        SPOT = SPOT + 1
C
C       read in record from second file
C
 105    CONTINUE
```

```
C     read a record from the second display file
      IF ((BENT) .OR. (REALT) .OR. (COS4)) THEN
       IF (ICONTR(SPOT) .EQ. 2HCA) THEN
        READ(11,2010,IOSTAT=ISTAT,ERR=994,END=107)
     1    (MSG1(I1),I1=1,52),CONTRY(1),CONTRY(2),IUSER,
     1    (TYPE(I1),I1=1,4)
       ELSE
        READ(11,2011,IOSTAT=ISTAT,ERR=994,END=107)
     1    (MSG1(I1),I1=1,52),CONTRY(1),CONTRY(2),IUSER,
     1    ELTI2,ELTI3,ELTI4
       ENDIF
      ELSE
       READ(11,1010,IOSTAT=ISTAT,ERR=994,END=107)(MSG1(I1),I1=1,52),
     1   ELTI1,ELTI2,ELTI3,ELTI4
      ENDIF
      IF (ISTAT .NE. 0) THEN
 994    CONTINUE
        WRITE(1,106)ISTAT
 106    FORMAT(' READ ERROR ON SECOND FILE, ISTAT IS ',I4)
        GO TO 999
      ENDIF
C     write a record onto the output logical unit
      IF ((BENT) .OR. (REALT) .OR. (COS4)) THEN
       IF (ICONTR(SPOT) .EQ. 2HCA) THEN
         WRITE(UNIT,2020)NLINE,(MSG1(I1),I1=1,52),
     1     CONTRY(1),CONTRY(2),IUSER,(TYPE(I1),I1=1,4)
       ELSE
         WRITE(UNIT,2021)NLINE,(MSG1(I1),I1=1,52),
     1     CONTRY(1),CONTRY(2),IUSER,ELTI2,ELTI3,ELTI4
       ENDIF
      ELSE
       WRITE(UNIT,1020)NLINE,(MSG1(I1),I1=1,52),ELTI1,ELTI2,ELTI3,
     1  ELTI4
      ENDIF
      LINE=LINE+1
      NLINE=NLINE+1
C
CC    if end of page, skip to next page
C
      IF (LINE .GE. PAGE) THEN
        RSPGE=PAGE -LINE
        DO 5 I1=1,RSPGE+8
          WRITE(UNIT,1)
 5      CONTINUE
        WRITE(UNIT,1007)
        WRITE(UNIT,1008)
        IF ((BENT) .OR.(REALT) .OR.(COS4))THEN
         WRITE(UNIT,2022)
        ELSE
         WRITE(UNIT,1009)
        ENDIF
        LINE=3
C       NLINE=1
      ENDIF
C
CC    SEE IF END OF QUERY
C
      IF (NLINE .EQ. POS) THEN
 22     CONTINUE
        CPOS = POS
        READ(14,FMT=1022,END=23)POS,FLAT,FLONG,FRADI
        IF (POS .EQ. CPOS) GO TO 22
        SPOT = SPOT + 1
      ENDIF
      GO TO 105
 23   CONTINUE
      POS = 999
      GO TO 105
     ENDIF
 107 CONTINUE
     IF (THIRD ) THEN
       RSPGE=PAGE-LINE
       DO 6 I1=1,RSPGE+8
        WRITE(UNIT,1)
 6     CONTINUE
       NLINE=1
       LINE=3
     ENDIF
C
CC   if the third output is desired then output its headers
C
     IF (THIRD) THEN
       IF (TST .EQ. 1) GO TO 108
       TST=1
       WRITE(UNIT,1011)
       WRITE(UNIT,1012)
       WRITE(UNIT,1013)
 108   CONTINUE
       REWIND(12)
C
CC     initialize the buffer
C
       I1=1
       CALL FILBS(408,240,MSG1,I1)
```

```
C
C       read in record from third file
C
  109   CONTINUE
C       read a record from the third output file
        READ(12,1014,IOSTAT=ISTAT,ERR=995,END=111)(MSG1(I1),I1=1,50)
        IF (ISTAT .NE. 0) THEN
 995     CONTINUE
         WRITE(1,110)ISTAT
 110     FORMAT(' READ ERROR ON THIRD FILE, ISTAT IS ',I4)
         GO TO 999
        ENDIF
C       write a record onto the output logical unit
        WRITE(UNIT,1021)NLINE,(MSG1(I1),I1=1,50)
        LINE=LINE+1
        NLINE=NLINE+1
C
C       if end of page then skip to the next page
C
        IF (LINE .GE. PAGE) THEN
          RSPGE=PAGE-LINE
          DO 7 I1=1,RSPGE+8
           WRITE(UNIT,1)
 7        CONTINUE
          WRITE(UNIT,1011)
          WRITE(UNIT,1012)
          WRITE(UNIT,1013)
          LINE=3
C         NLINE=1
        ENDIF
        GO TO 109
      ENDIF
  111 CONTINUE
      IF (FOURTH) THEN
        RSPGE=PAGE-LINE
        DO 8 I1=1,RSPGE+8
           WRITE(UNIT,1)
 8      CONTINUE
        LINE=3
        NLINE=1
      ENDIF
C
C     if the fourth output is desired then display its headers
C
      IF (FOURTH) THEN
        IF (FST .EQ. 1) GO TO 113
         FST=1
         WRITE(UNIT,1015)
         WRITE(UNIT,1016)
         WRITE(UNIT,1017)
  113   CONTINUE
        REWIND(13)
C
C       initialize buffer with blanks
C
        I1=1
        CALL FILBS(40B,240,MSG1,I1)
C
C       read in record from fourth file
C
  114   CONTINUE
C       read a record from the fourth output file
        READ(13,1018,IOSTAT=ISTAT,ERR=997,END=116)(MSG1(I1),I1=1,50)
        IF (ISTAT .NE. 0) THEN
  997     CONTINUE
          WRITE(1,115)ISTAT
  115     FORMAT(' READ ERROR ON FOURTH FILE, ISTAT IS ',I4)
          GO TO 999
        ENDIF
C       write a record from the output file
        WRITE(UNIT,1021)NLINE,(MSG1(I1),I1=1,50)
        LINE=LINE+1
        NLINE=NLINE+1
C
C       if end of page then skip to next page
C
        IF (LINE .GE. PAGE) THEN
         RSPGE=PAGE-LINE
         DO 9 I1=1,RSPGE+8
          WRITE(UNIT,1)
 9       CONTINUE
         WRITE(UNIT,1015)
         WRITE(UNIT,1016)
         WRITE(UNIT,1017)
         LINE=3
C        NLINE=1
        ENDIF
        GO TO 114
      ENDIF
```

```
  116 CONTINUE
C     IF (FOURTH) THEN
        RSPGE=PAGE-LINE
        DO 11 I1=1,RSPGE+8
          WRITE(UNIT,1)
 11     CONTINUE
        LINE=3
        MLINE=1
C     ENDIF
C
C     ******** OUTPUT IMAGE DATA ********
      IF (IMAGE) THEN
C
C       initialize variables
C
        PST=0
        SST=0
        TST=0
        FST=0
C
C     read from the header file
C
      REWIND (14)
      READ(14,FMT=1022,IOSTAT=ISTAT)POS,FLAT,FLONG,FRADI
      CPOS=POS
      CRADI=FRADI
      CLAT=FLAT
      CLONG=FLONG
 800  READ(14,FMT=1022,IOSTAT=ISTAT,END=721)POS,FLAT,FLONG,FRADI
      IF (CPOS .EQ. POS) THEN
         CPOS=POS
         CRADI=FRADI
         CLAT=FLAT
         CLONG=FLONG
         GO TO 800
      ENDIF
C
      IF (ISTAT .NE. 0) THEN
         WRITE(1,720)ISTAT
 720     FORMAT(' ERROR IN READING HEADER FILE, IOS IS ',I4)
         GO TO 999
      ENDIF
      GO TO 722
 721  POS=999
 722  CONTINUE
C
C
      PAGE=43
      IF (CRADI .EQ. 0.0) THEN
        LINE=7
      ELSE
        LINE=7
      ENDIF
      MLINE=1
C
      IF (PRIM) THEN
        P=3.141592654/180.
        R=6378.
        IF (PST .EQ. 1) GO TO 700
        PST=1
        WRITE(UNIT,1000)
        WRITE(UNIT,1001)
        WRITE(UNIT,1002)
        WRITE(UNIT,1)
        WRITE(UNIT,1023)CLAT,CLONG,CRADI
        WRITE(UNIT,1)
  700   CONTINUE
        REWIND(21,IOSTAT=ISTAT,ERR=1993)
C
C     initialize input buffer with blanks
C
        I1=1
        CALL FILBS(40B,240,MSG1,I1)
C
C     read in record from primary file
C
 701    CONTINUE
        READ(21,FMT=1004,IOSTAT=ISTAT,ERR=1993,END=703)
     1    (MSG1(I1),I1=1,50)
C
        IF (ISTAT .NE. 0) THEN
          WRITE(1,702)ISTAT
 702      FORMAT(' READ ERROR, ISTAT IS ',I4)
          GO TO 999
 1993     CONTINUE
          WRITE(1,1992)ISTAT
 1992     FORMAT(' ERROR IN READING FILE ',I4)
        ENDIF
        DECODE(9,1006,TRYA)LUTLA
        DECODE(9,1006,TRYB)LUTLG
```

```
C
C     compare to see if this is the correct record
C
      IF (MEVET .NE. LEVET) GO TO 500
      IF ((MPASS(1) .EQ. LPASS(1)) .AND. (MPASS(2) .EQ. LPASS(2))
     1    .AND. (MPASS(3) .EQ. LPASS(3)) .AND.
     1    (MPASS(4) .EQ. LPASS(4))) GO TO 502
      GO TO 500
C
C     have the correct record
C
  502 CONTINUE
      IF ((ABS(LUTLA-LATA) .LT. .001) .AND.
     1    (ABS(LUTLG-LONGA) .LT. .001)) THEN
          I1=64
          CALL MOVBS(MBUF,55,2,MSG1,I1)
          MSG1(34)=2HA
          I1=70
          CALL FPNBS(LATB,177744B,MSG1,I1)
          I1=81
          CALL FPNBS(LONGB,177744B,MSG1,I1)
          I1=92
          CALL MOVBS(MBUF,57,2,MSG1,I1)
          MSG1(48)=2HB
       ELSE
          I1=64
          CALL MOVBS(MBUF,57,2,MSG1,I1)
          MSG1(34)=2HB
          I1=70
          CALL FPNBS(LATA,177744B,MSG1,I1)
          I1=81
          CALL FPNBS(LONGA,177744B,MSG1,I1)
          I1=92
          CALL MOVBS(MBUF,55,2,MSG1,I1)
          MSG1(48)=2HA
       ENDIF
  550  CONTINUE
C
C     date
C
      I1=2
      CALL MOVBS(LBUF,9,6,MSG1,I1)
C
C     satpas & id
C
      I1=I1+1
      CALL MOVBS(LBUF,19,8,MSG1,I1)
C
C     Mccref
C
      CALL FORM(10,LBUF(4),MSG1,I1,6)
C
C     event
C
      CALL FORM(10,LBUF(3),MSG1,I1,6)
C
C     Messnt
C
      I1=I1+2
      CALL FORM(10,LBUF(63),MSG1,I1,6)
C     frequency
      CALL FORM(10,LBUF(25),MSG1,I1,4)
C
C     elt lat and elt long
C
      I1=42
      CALL FPNBS(LUTLA,177744B,MSG1,I1)
      I1=53
      CALL FPNBS(LUTLG,177744B,MSG1,I1)
C
      IF (OTHDAT) THEN
        WRITE(21,1004)(MSG1(I1),I1=1,50)
      ELSE
        WRITE(10,1004)(MSG1(I1),I1=1,50)
        POSN=POSN+1
      ENDIF
 1000 FORMAT('                    PRIMARY DATA')
 1001 FORMAT('                                                  PRIMARY',
     1       '                    SECONDARY')
 1002 FORMAT('   DATE  SATPAS  MCCREF  EVENT  MESSNT     ELTLAT',
     1       '     ELTLONG          ELTLAT    ELTLONG',
     1       '       DIFF      DLAT      DLONG')
 1004 FORMAT(50A2)
 1005 FORMAT(' ***ERROR IN MCCELF, ISTAT IS ',I4)
      RETURN
      END
```

```
      INTEGER KEYS(8,8), PROM(8), RK
      INTEGER FLG, AA, IFBRK
C
C       FLG   - flag to indicate if /E used
C       AA    - dummy parameter for IFBRK
C
C
      COMMON/SELCT/ALLUT,ALSAT,SAT,SX,CX,SPESAT,PRIM,SECOND,THIRD,
     1       FOURTH,PST,SST,TST,FST,UNIT,DIF,BENT,REALT,COS4,CBC12,
     1       CBC24,ALLFR,POSN,START,EXIT,TAPE,SPECIF,SEVENT,PASS,
     1       IMAGE,CONTRY,IUSER,TYPE,ELTANG
      INTEGER CONTRY(3),IUSER,TYPE(4)
      REAL ELTANG
      LOGICAL ALLUT,ALSAT,SAT,SX,CX,PRIM,SECOND,THIRD,FOURTH
      LOGICAL BENT,REALT,COS4,CBC12,CBC24,ALLFR,EXIT,TAPE,SPECIF
      LOGICAL IMAGE,OTHDAT
      INTEGER SPESAT(5),I,PST,SST,TST,FST,MSG1(128),POSN,START
      INTEGER SEVENT,PASS(4)
      REAL DIF
C
C       ALLUT -flag to indicate if all locations wanted
C       ALSAT -flag to indicate if all satellites wanted
C       SAT   -flag to indicate if specific satellites wanted
C       SX    -flag to indicate if all SARSAT satellites wanted
C       CX    -flag to indicate if all COSPAS satellites wanted
C       SPESAT-array that contains the specific satellites desired
C
      DOUBLE PRECISION LATC,LATT,LONGT,LONGC,FA,DIFFR,DLAT,DLONG
      REAL R,P
C
C       LATC  -calculated latitude from LUT file
C       LATT  -true latitude as input by the user
C       LONGT -true longitude as input by the user
C       LONGC -calculated longitude from the LUT file
C       DLAT  -difference in latitude in km.
C       DLONG -difference in longitude in km.
C
      INTEGER TRYA(3),TRYB(3)
      INTEGER MPASS(4),MEVET,LPASS(4),LEVET,ITEM(3),MREF,MSG2(80)
      REAL LONGA,LONGB,LATA,LATB
      EQUIVALENCE(LATA,MBUF(32))
      EQUIVALENCE(LONGA,MBUF(34))
      EQUIVALENCE(LATB,MBUF(43))
      EQUIVALENCE(LONGB,MBUF(45))
      EQUIVALENCE(MPASS(1),MBUF(10))
      EQUIVALENCE(MEVET,MBUF(3))
      EQUIVALENCE(LPASS(1),LBUF(10))
      EQUIVALENCE(LEVET,LBUF(3))
      EQUIVALENCE(MREF,LBUF(4))
      EQUIVALENCE(LUTLA,LBUF(28))
      EQUIVALENCE(LUTLG,LBUF(30))
      EQUIVALENCE(TRYA(1),MSG1(54))
      EQUIVALENCE(TRYB(1),MSG1(59))
C       KEYS    - array containing contents of soft keys
C       PROM    - array of values to return for soft keys
C       RK      - actual value returned
       DATA ITEM/6HMCCREF/
C
C*****************************************************************************
C
C                EXECUTABLE  CODE
C
C*****************************************************************************
C
      ISYS = LOGLU(SYS)
C
  100 CONTINUE
C
      I1=1
      CALL FILBS(40B,240,MSG1,I1)
C
C     determine if there is an MCCREF number linked in the LUT record
C
      IF (MREF .EQ. 0) GO TO 550
C
C     initialize the chain to the MCC detail records
C
      CALL DQFND(IBASE,IDMCC,ITEM,MREF,1,ISTAT)
      IF (ISTAT .EQ. 0) GO TO 500
      IF (ISTAT .EQ. 155) GO TO 550
      WRITE(LU,1005)ISTAT(1)
C
C     read the MCC record
C
  500 CONTINUE
      CALL DQGET(IBASE,IDMCC,5,MBUF,ISTAT)
      IF (ISTAT .EQ. 0) GO TO 501
      IF (ISTAT .EQ. 155) GO TO 550
      WRITE(LU,1005)ISTAT(1)
  501 CONTINUE
```

```
FTN4
$FILES(15,15)
      SUBROUTINE PRIMR(OTHDAT)
     -          SEF-40-00  V00  (840725.1427)
      IMPLICIT NONE
C                                                                       *
C-----------------------------------------------------------------------*
C                                                                       *
C  SARSAT EVALUATION FACILITY                                           *
C                                                                       *
C-----------------------------------------------------------------------*
C                                                                       *
C    DATE     VERSION        DESCRIPTION           AUTHOR              *
C  83/10/05     00              -----             SUZANNE Y. SLINN     *
C                                                                       *
C-----------------------------------------------------------------------*
C                                                                       *
C DESCRIPTION:                                                          *
C   This subroutine displays the first of the four output files on the  *
C desired logical unit.                                                 *
C                                                                       *
C CALLING SEQUENCE:                                                     *
C   CALLED FROM: ALOCS                                                  *
C                                                                       *
C       CALL PRIMR                                                      *
C-----------------------------------------------------------------------*
C
C-----------------------------------------------------------------------*
C                                                                       *
C                        DATA DECLARATIONS                              *
C                                                                       *
C-----------------------------------------------------------------------*
C
C
      COMMON /LOGG/ LODAT(3), HIDAT(3), STDAT, ENDAT,OUTLU(5),RANGE
      INTEGER LODAT, HIDAT, OUTLU, RANGE
      DOUBLE PRECISION STDAT, ENDAT
C
      REAL LOCLAT, LOCLNG, RADIUS,LUTLA,LUTLG
C
C         LOCLAT - latitude of specified location
C         LOCLNG - longitude of specified location
C         RADIUS - maximum distance between specified location and
C                  file locations
C
      COMMON /LOCN/ LOCLAT, LOCLNG, RADIUS
C
      INTEGER IBASE(5), ILEVL(3), IMODE, ISTAT(10), LIST
      INTEGER ITDTE(3), ITTST(3), ITSAR(3), ITMCC(3), ITSPD(3)
      INTEGER IDLUT(3), IDTST(3), IDSAR(3), IDMCC(3)
C
      INTEGER SBUF(68), LBUF(98), MBUF(76), FBUF(50) ,BGBUF(500)
C
      COMMON /BASE/ IBASE, ILEVL, IMODE, ISTAT, LIST, ITDTE,
     -              ITTST, ITSAR, ITMCC, ITSPD, IDLUT, IDMCC,
     -              IDSAR, IDTST, SBUF, LBUF, MBUF, FBUF
C
      DOUBLE PRECISION QTIME, SDT
C
C         SDT    - start date
C
C
C         LODAT  - ascii start date YYMMDD
C         STDAT  - low date in seconds from 1980
C         ENDAT  - high date in seconds from 1980
C
C         OUTLU  - output device, LU or filename
C
      INTEGER RBUF(40), IPARM(5), PBUF(33)
      INTEGER LU, RET
      INTEGER SYS, ISYS, LOGLU
C
C         RBUF   - receiving buffer
C         IPARM  - return buffer for RMPAR call
C         PBUF   - return buffer from PARSE
C
      REAL ONEDAY
      INTEGER DATCH(3), LOOP, DUM(7), I1
      INTEGER UNIT
C
C
      LOGICAL LUT, MCC, SAR, TEST, ALL
C
C         LUT    - flag to indicate LUT location detail file selected
C         MCC    - flag to indicate MCC location detail file selected
C         SAR    - flag to indicate SARSTAT incident detail file selected
C         TEST   - flag to indicate FIELD test master file selectedlected
C         ALL    - all files will be searched
C
```

```
C         FORMATS
 1001 FORMAT ("IMAGE ERROR - (SELUT) - ",I5)
 1002 FORMAT (" $LUT$$ SAT ID &       RCC/SAR   MCC   FIELD ")
 1003 FORMAT ("  DATE  PASS NO  EVENT INCIDENT  REF#  TEST      LATITUDE
     -  LONGITUDE DISTANCE")
 1004 FORMAT (38A2)
 1005 FORMAT (' There are ',I4, ' hits.')
 1006 FORMAT(' Do you wish a specific countrys beacon? ) ')
 1007 FORMAT(A1)
 1008 FORMAT(' Which country do you wish?')
 1009 FORMAT(3A2)
 1010 FORMAT(' Which Canadian beacon(s) do you want?')
 1011 FORMAT(11(4A2,1X))
  999 CONTINUE
C     CALL DQCLS(IBASE,6HSARIF ,1,ISTAT)
      OPEN(30,FILE='SCRATCH',STATUS='OLD')
      IF (SPECON .EQ. 1HY) THEN
       WRITE(30,1023)POSN,START,CONTRY(1)
      ELSE
       WRITE(30,1023)POSN,START,TYPE(1)
      ENDIF
 1023 FORMAT(I5,1X,I5,1X,A2)
      CLOSE(30)
      CLOSE(10)
      CLOSE(11)
      CLOSE(12)
      CLOSE(13)
      CLOSE(14)
      CLOSE(21)
      CLOSE(22)
      CLOSE(23)
      CLOSE(24)
      CLOSE(UNIT)
      CALL EXEC(6)
      STOP
      END



C
C*************************************************************************
C
C Subroutine to break a single precision integer into its bit string.
C
C*************************************************************************
C
      SUBROUTINE BITSC(IVAL,IBIT)
C
      INTEGER IVAL,IBIT(16),IB,I,J
C
      IBIT(16) = 0
      IB = IVAL
      IF (IB .GE. 0 ) GO TO 5
        IBIT(16) = 1
        IB = 32768 + IB
C
 5      DO 10 I = 1,15
            J = 16 - I
            IBIT(J) = 0
            IF ((IB - 2**(J-1)) .LT. 0) GO TO 10
            IBIT(J) = 1
            IB = IB -2**(J-1)
 10     CONTINUE
C
        RETURN
        END
```

```
                    ENDIF
                  ENDIF
                ENDIF
              ENDIF
            ENDIF
          ENDIF
        ENDIF
      ENDIF
 1      CONTINUE
        GO TO 200
        ENDIF
      ENDIF
 2    CONTINUE
C
C     Elt angle calculation
C
      IF ((LBUF(10) .EQ. 2HS1) .OR. (LBUF(10) .EQ. 2HS2))H=850.
      A = ABS(CTA) * PHI/180.
      R1 = AE + H
C
C     ELTANG=SQRT((AE*SIN(A)*SIN(A))/(AE+2*(AE+H)*(1-COS(A))))
      ELTANG=(AE*SIN(A))/SQRT((2*AE*(AE+H)*(1-COS(A)))+(H*H))
      ELTANG = 90-((ABS(CTA))+(ASIN(ELTANG)*180 /PHI))
C
      ENDIF
C
C     call the output routines according to type of output desired
C
      OTHDAT= FALSE
      IF (PRIM) CALL PRIMR(OTHDAT)
      IF (SECOND) CALL SECDY(OTHDAT,SPECON)
      IF (THIRD) CALL TERTI(OTHDAT)
      IF (FOURTH) CALL QUADY(OTHDAT)
      IF (IMAGE) THEN
       OTHDAT= TRUE.
C
C      Obtain chain info
C
       IMODE=401
       CALL DQINF(IBASE,IDLUT,IMODE,ISTAT,INFO,14)
       IF (ISTAT(1) .EQ. 0) GO TO 124
       WRITE(1,1001)ISTAT(1)
       GO TO 999
 124   CONTINUE
C
C      Get the alternate solution
C
       IMODE=4
       CALL DQGET(IBASE,IDLUT,IMODE,LBUF,ISTAT,ALTSOL,2)
       IF (ISTAT(1) .EQ. 0) GO TO 500
       WRITE(LU,1001)ISTAT(1)
       GO TO 999
 500   CONTINUE
C
C      Write out image data
C
       IF (PRIM) CALL PRIMR(OTHDAT)
       IF (SECOND) CALL SECDY(OTHDAT,SPECON)
       IF (THIRD) CALL TERTI(OTHDAT)
       IF (FOURTH) CALL QUADY(OTHDAT)
C
C      Get the original record back
C
       CALL DQGET(IBASE,IDLUT,IMODE,LBUF,ISTAT,ALTSOL,2)
       IF (ISTAT(1) .EQ. 0) GO TO 125
       WRITE'LU,1001) ISTAT(1)
       GO TO 999
 125   CONTINUE
C
C      Restore the chain information
C
       IMODE=402
       CALL DQINF(IBASE,IDLUT,IMODE,ISTAT,INFO,14)
       IF (ISTAT(1) .EQ. 0) GO TO 126
       WRITE(LU,1001)ISTAT(1)
       GO TO 999
 126   CONTINUE
      ENDIF
 510  CONTINUE
      OEVENT = LBUF(3)
      OLDSAT(1)=LBUF(10)
      OLDSAT(2)=LBUF(11)
      OLDSAT(3)=LBUF(12)
      OLDSAT(4)=LBUF(13)
      GO TO 200
  100 CONTINUE
      WRITE(LU,1005)(POSN-START)
```

```
IF (LBUF(24) .EQ. -21283) TYPE(4) = 2HB1
IF (LBUF(24) .EQ. -21287) TYPE(4) = 2HB2
IF (LBUF(24) .EQ. -21296) TYPE(4) = 2HB3
IF (LBUF(24) .EQ. -21302) TYPE(4) = 2HB4
IF (LBUF(24) .EQ. -21311) TYPE(4) = 2HB5
IF (LBUF(24) .EQ. -14243) TYPE(4) = 2HT1
IF (LBUF(24) .EQ. -14247) TYPE(4) = 2HT2
IF (LBUF(24) .EQ. -14256) TYPE(4) = 2HT3
IF (LBUF(24) .EQ. -14262) TYPE(4) = 2HT4
IF (LBUF(24) .EQ. -14271) TYPE(4) = 2HT5
IF (LBUF(24) .EQ. -14251) TYPE(4) = 2HT6
GO TO 2
ENDIF

C
DO 1 I=1,11
 IF (BEACON(I,1) .EQ. 2HCG) THEN
  TYPE(1) = 2HCG
  TYPE(2) = 2HEP
  TYPE(3) = 2HIR
  IF (BEACON(I,4) .EQ. 2HB1) THEN
   IF (LBUF(24) .EQ. -21283) THEN
    TYPE(4) =2HB1
    GO TO 2
   ENDIF
  ELSE
   IF (BEACON(I,4) .EQ. 2HB2) THEN
    IF (LBUF(24) .EQ. -21287) THEN
      TYPE(4) = 2HB2
      GO TO 2
    ENDIF
   ELSE
    IF (BEACON(I,4) .EQ. 2HB3) THEN
     IF (LBUF(24) .EQ. -21296) THEN
       TYPE(4) = 2HB3
       GO TO 2
     ENDIF
    ELSE
     IF (BEACON(I,4) .EQ. 2HB4) THEN
      IF (LBUF(24) .EQ. -21302) THEN
        TYPE(4) = 2HB4
        GO TO 2
      ENDIF
     ELSE
      IF (BEACON(I,4) .EQ. 2HB5) THEN
       IF (LBUF(24) .EQ. -21311) THEN
         TYPE(4) = 2HB5
         GO TO 2
       ENDIF
      ENDIF
     ENDIF
    ENDIF
   ENDIF
 ENDIF
ENDIF
IF (BEACON(I,1) .EQ. 2HDN) THEN
 TYPE(1) = 2HDN
 TYPE(2) = 2HDU
 TYPE(3) = 2HNI
 IF (BEACON(I,4) .EQ. 2HT1) THEN
  IF (LBUF(24) .EQ. -14243) THEN
    TYPE(4) = 2HT1
    GO TO 2
  ENDIF
 ELSE
  IF (BEACON(I,4) .EQ. 2HT2) THEN
   IF (LBUF(24) .EQ. -14247) THEN
    TYPE(4) = 2HT2
    GO TO 2
   ENDIF
  ELSE
   IF (BEACON(I,4) .EQ. 2HT3) THEN
    IF (LBUF(24) .EQ. -14256) THEN
      TYPE(4) = 2HT3
      GO TO 2
    ENDIF
   ELSE
    IF (BEACON(I,4) .EQ. 2HT4) THEN
     IF (LBUF(24) .EQ. -14262) THEN
       TYPE(4) = 2HT4
       GO TO 2
     ENDIF
    ELSE
     IF (BEACON(I,4) .EQ. 2HT5) THEN
      IF (LBUF(24) .EQ. -14271) THEN
        TYPE(4) = 2HT5
        GO TO 2
      ENDIF
     ELSE
      IF (BEACON(I,4) .EQ. 2HT6) THEN
       IF (LBUF(24) .EQ. -14251) THEN
         TYPE(4) = 2HT6
         GO TO 2
```

```
C
C     Check to see if 406 data is desired
C
      IF ((BENT) .OR. (REALT) .OR. (COS4)) THEN
C
C           Check that have correct country/beacon
C
            CALL BITSG(LBUF(21),IBIT)
            ICOUNR = IBIT(1)+2*IBIT(2)+4*IBIT(3)+8*IBIT(4)
     1             +16*IBIT(5)+32*IBIT(6)+64*IBIT(7)+128*IBIT(8)
            IUSER=IBIT(9)+2*IBIT(10)+4*IBIT(11)+8*IBIT(12)
C
C           IF A SPECIFIC COUNTRY IS NOT WANTED DETERMINE THE COUNTRY
C           OF EACH BEACON
C
            IF (SPECON .NE. 1HY) THEN
             IF (ICOUNR .EQ. 111) THEN
               CONTRY(1) = 2HUS
               CONTRY(2) = 2HA
             ENDIF
             IF (ICOUNR .EQ. 121) THEN
               CONTRY(1) = 2HCA
               CONTRY(2) = 2HNA
             ENDIF
             IF (ICOUNR .EQ. 211) THEN
               CONTRY(1) = 2HFR
               CONTRY(2) = 2HAN
             ENDIF
             IF (ICOUNR .EQ. 221) THEN
               CONTRY(1) = 2HUS
               CONTRY(2) = 2HSR
             ENDIF
             IF (ICOUNR .EQ. 231) THEN
               CONTRY(1) = 2HNO
               CONTRY(2) = 2HRW
             ENDIF
             IF (ICOUNR .EQ. 241) THEN
               CONTRY(1) = 2HUK
               CONTRY(2) = 2H
             ENDIF
            ENDIF
C
            IF (SPECON .EQ. 1HY) THEN
              IF ((CONTRY(1).EQ.2HUS).AND.(CONTRY(2).EQ.2HA ))THEN
               IF (ICOUNR .NE. 111) GO TO 200
              ELSE
               IF (CONTRY(1) .EQ. 2HCA) THEN
                IF (ICOUNR .NE. 121) GO TO 200
               ELSE
                IF (CONTRY(1) .EQ. 2HFR) THEN
                 IF (ICOUNR .NE. 211) GO TO 200
                ELSE
                 IF((CONTRY(1) .EQ. 2HUS) AND.(CONTRY(2).EQ.2HSR))THEN
                  IF (ICOUNR .NE. 221) GO TO 200
                 ELSE
                  IF (CONTRY(1) .EQ. 2HNO) THEN
                   IF (ICOUNR .NE. 231) GO TO 200
                  ELSE
                   IF(CONTRY(1) .EQ. 2HUK) THEN
                    IF (ICOUNR .NE. 241) GO TO 200
                   ENDIF
                  ENDIF
                 ENDIF
                ENDIF
               ENDIF
              ENDIF
C
C             Correct country code, if CANADA, determine if correct beacon
C             IF (CONTRY(1) .EQ. 2HCA) THEN
C
C              SEE IF ALL CANADIAN BEACONS ARE DESIRED
C
               IF (BEACON(1,1) .EQ. 2H  ) THEN
                 IF ((LBUF(24) .EQ. -21283).OR.(LBUF(24) .EQ. -21287)
     1           .OR.(LBUF(24) .EQ. -21296).OR.(LBUF(24) .EQ. -21302)
     1           .OR.(LBUF(24) .EQ. -21311)) THEN
                    TYPE(1) = 2HCG
                    TYPE(2) = 2HEP
                    TYPE(3) = 2HIR
               ELSE
                    TYPE(1) = 2HDN
                    TYPE(2) = 2HDU
                    TYPE(3) = 2HNI
               ENDIF
```

```
C
C     determine if all COSPAS satellites are desired
C
      IF (CX) THEN
         IF ((LBUF(10) .NE. 2HC1) .AND. (LBUF(10) .NE. 2HC2) .AND.
     1      (LBUF(10) .NE. 2HC3) .AND. (LBUF(10) .NE. 2HC4)) GO TO 200
      ENDIF
C
C     determine if a specific satellite is desired
C
      IF (SAT) THEN
         IF ((LBUF(10) .NE. SPESAT(1)) .AND. (LBUF(10) .NE.SPESAT(2))
     1    .AND. (LBUF(10) .NE. SPESAT(3))
     1    .AND. (LBUF(10) .NE. SPESAT(4))
     1    .AND. (LBUF(10) .NE. SPESAT(5))) GO TO 200
      ENDIF
C
C     determine if a specific SATPAS is desired
C
      IF (SPECIF) THEN
        IF ((LBUF(10) .NE. PASS(1)) .AND.
     1      (LBUF(11) .NE. PASS(2)) .AND.
     1      (LBUF(12) .NE. PASS(3)) .AND.
     1      (LBUF(13) .NE. PASS(4))) GO TO 200
      ENDIF
C
C     determine which frequency of ELT is desired for output
C
      CHOSE=.FALSE.
C
C     if not all the ELT frequencys are desired then
C
      IF (.NOT. (ALLFR)) THEN
C
C         if CBC121 and CBC243 are desired then
C
          IF (CBC12 .AND. CBC24) THEN
           IF ((LBUF(25) .NE. 8) .AND.(LBUF(25) .NE. 16)) GO TO 200
           CHOSE = .TRUE.
           GO TO 111
          ENDIF
C
C         if CBC121 is desired then
C
          IF (CBC12) THEN
            IF (LBUF(25) .NE. 8) GO TO 200
            CHOSE=.TRUE.
          ENDIF
C         if CBC243 is desired then
C
          IF (CBC24) THEN
           IF (LBUF(25) .NE. 16) GO TO 200
           CHOSE = .TRUE.
          ENDIF
C
C         if 406 Kb/s real time is desired then
C
          IF (REALT) THEN
           IF (LBUF(25) .NE. 2) GO TO 200
           CHOSE = .TRUE.
          ENDIF
C
C         if 406 'bent pipe' data is desired then
C
          IF (BENT) THEN
           IF (LBUF(25) .NE. 1) GO TO 200
           CHOSE= .TRUE.
          ENDIF
C
C         if 406 2.4Kb/s COSPAS stored data is desired then
C
          IF (COS4) THEN
           IF (LBUF(25) .NE. 4) GO TO 200
           CHOSE=.TRUE.
          ENDIF
C
C         else an invalid option was chosen
C
          IF ( NOT.(CHOSE))THEN
             WRITE(LU,564)
  564        FORMAT(' AN ILLEGAL FREQUENCY CHOICE WAS MADE. DO AGAIN')
             GO TO 999
          ENDIF
        ENDIF
  111 CONTINUE
```

```
C
C     Open database
C
      CALL DQOPN(IBASE,ILEVL,1,ISTAT)
      IF (ISTAT(1) .NE. 0)WRITE(1,1021)ISTAT(1)
 1021 FORMAT(' ERROR IN OPENING DATABASE IN ALOCS ',I4)
C
      LU = 1
      ONEDAY = -24.0
      UNIT = 99
      PST = 0
      SST = 0
      TST = 0
      FST = 0
C
C     DETERMINE IF A SPECIFIC COUNTRY'S 406 BEACON IS DESIRED
C
      IF ((BENT) .OR. (REALT) .OR. (COS4)) THEN
      WRITE(LU,1006)
      READ(LU,1007)SPECON
      IF (SPECON .EQ. 1HY) THEN
        WRITE(LU,1008)
        READ(LU,1009)(CONTRY(I),I=1,3)
C
C       IF CANADIAN BEACONS, DETERMINE WHICH ID'S ARE DESIRED
C
        IF (CONTRY(1) .EQ. 2HCA) THEN
            WRITE(LU,1010)
            READ(LU,1011)((BEACON(I,J),J=1,4),I=1,11)
        ENDIF
       ENDIF
      ENDIF
C
C     set up the loop counter for the start and end dates
C
      DO 100 LOOP = 1,RANGE
      I1 = 1
      ONEDAY = ONEDAY + 24.0
      SDT = QTIME(LODAT,I1,3,ONEDAY)
      I1 = 1
      CALL DTIMA(SDT,DUM,I1)
      I1 = 1
      CALL MOVBS(DUM,1,6,DATE1,I1)
C
C         initialize chain to LUT detail records
C
      IF (SPECIF) THEN
        CALL DQFND(IBASE,IDLUT,ITSPD,PASS,4,ISTAT)
        IF (ISTAT(1) .EQ. 0) GO TO 200
        IF (ISTAT(1) .EQ. 107 .OR. ISTAT(1) .EQ. 156) GO TO 100
        WRITE(LU,1001)ISTAT(1)
      ELSE
        CALL DQFND(IBASE,IDLUT,ITDTE,DATE1,3,ISTAT)
        IF (ISTAT(1) .EQ. 0)  GO TO 200
        IF (ISTAT(1) .EQ. 107 OR. ISTAT(1) .EQ. 156)  GO TO 100
        WRITE(LU,1001) ISTAT(1)
      ENDIF
C
C         read the LUT records
C
  200 CONTINUE
      CALL DQGET(IBASE,IDLUT,5,LBUF,ISTAT)
      IF (ISTAT(1) .EQ. 0)  GO TO 300
      IF (ISTAT(1) .EQ. 155)  GO TO 100
      WRITE(LU,1001) ISTAT(1)
C
C         process the record
C
  300 CONTINUE
      IF ((OEVENT .EQ. LBUF(3)) .AND. (OLDSAT(1) .EQ. LBUF(10))
     1 .AND. (OLDSAT(2) .EQ. LBUF(11)) .AND.
     1 (OLDSAT(3) .EQ. LBUF(12)) .AND. (OLDSAT(4) .EQ. LBUF(13)))
     1 GO TO 200
C
C     if not all the locations desired then
C
      IF (.NOT.(ALLUT)) THEN
C
C        determine if this records location is in the desired radius
C
         DIF = DSTNC(LOCLNG,LOCLAT,LUTLG,LUTLA)
         IF (DIF .GT. RADIUS) GO TO 200
      ENDIF
C
C     determine if all SARSAT satellites are desired
C
      IF (SX) THEN
         IF ((LBUF(10) .NE. 2HS1) .AND. (LBUF(10) .NE. 2HS2) .AND.
     1      (LBUF(10) .NE. 2HS3) .AND. (LBUF(10) .NE. 2HS4))GO TO 200
      ENDIF
```

```
FTN4
$FILES(15,15)
      PROGRAM ALOCS
C     -            SEF-40-00  V00  (840731.0806)
      IMPLICIT NONE
C
      COMMON /LOGG/ LODAT(3), HIDAT(3), STDAT, ENDAT,OUTLU(5),RANGE
      INTEGER LODAT, HIDAT, OUTLU, RANGE
      DOUBLE PRECISION STDAT, ENDAT
C
      REAL LOCLAT, LOCLNG, RADIUS,H,N
C
      COMMON /LOCN/ LOCLAT, LOCLNG, RADIUS
C
      INTEGER IBASE(5), ILEVL(3), IMODE, ISTAT(10), LIST,ILIST(7)
      INTEGER ITDTE(3), ITTST(3), ITSAR(3), ITMCC(3), ITSPD(3)
      INTEGER IDLUT(3), IDTST(3), IDSAR(3), IDMCC(3)
C
      INTEGER SBUF(68), LBUF(98), MBUF(76), FBUF(50)
C
      COMMON /BASE/ IBASE, ILEVL, IMODE, ISTAT, LIST, ITDTE,
     -              IITST, ITSAR, ITMCC, ITSPD, IDLUT, IDMCC,
     -              IDSAR, IDTST, SBUF, LBUF, MBUF, FBUF
C
C
C
      REAL ONEDAY, LUTLA, LUTLG, DIF, DSTNC
C
C         ONEDAY - counter of 24 hours
C         LUTLA  - latitude on LUTELF record
C         LUTLG  - longitude on LUTELF record
C         DIF    - distance calculated
C
      DOUBLE PRECISION QTIME, SDT
      INTEGER LOOP, I, I1, DUM(7), DATE1(3)
C
C         LOOP   - variable for do loop
C         I,I1   - counters
C         DUM    - dummy buffer for call to QTIME
C         DATE1  - ascii date
C
      INTEGER LU, FST, MSG1(40), UNIT
      INTEGER*4 ALTSOL,INFO(7)
C
C         LU     - terminal LU set to 1
C         FST    - first time flag
C         MSG1   - output line buffer
C         UNIT   - fortran LU set to 99
C
      COMMON/SELCT/ALLUT,ALSAT,SAT,SX,CX,SPESAT,PRIM,SECOND,THIRD,
     1  FOURTH,PST,SST,TST,FST,UNIT,DIF,BENT,REALT,COS4,CBC12,
     1  CBC24,ALLFR,POSN,START,EXIT,TAPE,SPECIF,SEVENT,PASS,IMAGE,
     1  CONTRY,IUSER,TYPE,ELTANG
      LOGICAL ALLUT,ALSAT,SAT,SX,CX,PRIM,SECOND,THIRD,FOURTH,SPECIF
      LOGICAL BENT,REALT,COS4,CBC12,CBC24,ALLFR,CHOSE,EXIT,TAPE
      LOGICAL IMAGE,OTHDAT
      INTEGER SPESAT(5),I,PST,SST,TST,POSN,START,SEVENT,PASS(4)
      INTEGER PLACE, CONTRY(3), BEACON(11,4),SPECON,IBIT(16)
      INTEGER REGION,CASENO, ICOUNR, IUSER, J, TYPE(4)
      REAL ELTANG , PHI,AE,H,A,R1,CTA
      INTEGER OEVENT,OLDSAT(4)
C
      EQUIVALENCE (LUTLA,LBUF(28))
      EQUIVALENCE (LUTLG,LBUF(30))
      EQUIVALENCE (ALTSOL,LBUF(64))
      EQUIVALENCE (CTA,LBUF(34))
C
C           initialize variables
C
      PHI = 3.14159265
      AE = 6378.145
      H = 1000.
C
C
      OPEN(30,FILE='SCRATCH',STATUS='OLD')
      READ (30,1000)ALLUT,ALSAT,SAT,SX,CX,(SPESAT(I),I=1,5),PRIM,
     1    SECOND,THIRD,FOURTH
 1000 FORMAT(5(L1,1X),5A2,1X,4(L1,1X))
      READ (30,1019)PST,SST,TST,FST,UNIT,DIF,BENT,REALT,COS4,CBC12,
     1 CBC24,ALLFR,POSN,START,EXIT,TAPE,SPECIF,SEVENT,(PASS(I),I=1,4)
 1019 FORMAT(5(I5,1X),F10.3,1X,6(L1,1X),2(I5,1X),3(L1,1X),I5,1X,4A2)
      READ  (30,1020)(LODAT(I),I=1,3),(HIDAT(I),I=1,3),STDAT,ENDAT,
     1      (OUTLU(I),I=1,5),RANGE,LOCLAT,LOCLNG,RADIUS ,IMAGE,
     1      REGION,CASENO,PLACE,(IBASE(I),I=1,5)
 1020 FORMAT(3A2,1X,3A2,1X,F10.3,1X,F10.3,1X,5A2,1X,I8,1X,3(F10.3,1X),
     1 L1,A2,1X,I4,1X,I3,1X,5A2)
      CLOSE(30)
C
C     Open the output file
C
      CALL OUTP2(PLACE)
```

```
C
C       if end of page then skip to next page
C
        IF (LINE .GE. PAGE) THEN
         RSPGE=PAGE-LINE
         DO 99 I1=1,RSPGE+8
          WRITE(UNIT,1)
 99       CONTINUE
         WRITE(UNIT,1015)
         WRITE(UNIT,1016)
         WRITE(UNIT,1017)
         LINE=3
C        NLINE=1
        ENDIF
        GO TO 714
      ENDIF
  716 CONTINUE
C     IF (FOURTH) THEN
        RSPGE=PAGE-LINE
        DO 21 I1=1,RSPGE+8
          WRITE(UNIT,1)
 21     CONTINUE
        LINE=3
        NLINE=1
C     ENDIF
      ENDIF
 1000 FORMAT('                        PRIMARY DATA')
 1001 FORMAT('                                                    ',
     1       '     PRIMARY                    SECONDARY')
 1002 FORMAT('                                                    ',
     1       '   DATE  SATPAS  MCCREF  EVENT  MESSNT     ELTLAT',
     1       '     ELTLONG          ELTLAT    ELTLONG            ',
     1       ' DIFF      DLAT       DLONG')
 1004 FORMAT(48A2,3(1X,F10.4))
 1006 FORMAT(F10.3)
 1007 FORMAT('                         SECOND OUTPUT')
 1008 FORMAT('                        PRIMARY LOCATIOIN')
 1009 FORMAT('                                                    ',
     1       '     CTA     POINTS   SDEV      TREND    QUAL   PROB',
     1       '   NMWLS   TCA            QTIME       LOSTIM ',
     1       '   BIAS              CORR SCORE')
 1010 FORMAT(52A2,4(1X,I8))
 1011 FORMAT('                          THIRD OUTPUT')
 1012 FORMAT('                        PRIMARY LOCATION')
 1013 FORMAT('                                                    ',
     1       '     DRIFT     CTAI     TCAI      MAJAX     MINAX     ',
     1       '   AMEAN      BIASI')
 1014 FORMAT(50A2)
 1015 FORMAT('                         FOURTH OUTPUT')
 1016 FORMAT('                        PRIMARY LOCATION')
 1017 FORMAT('                                                    ',
     1       '     VARCTA    VARTCA    VARBIA     VARDI     CORCT   ',
     1       '  CORCB     CORCD     CORTB     CORTD     CORBD')
 1018 FORMAT(50A2)
 1019 FORMAT(1X,I3,')',3X,48A2,3(1X,F10.4))
 1020 FORMAT(1X,I3,')',3X,52A2,4(1X,I8))
 1021 FORMAT(1X,I3,')',3X,50A2)
 1022 FORMAT(I4,1X,F10.4,1X,F10.4,1X,F9.1)
 1023 FORMAT(' LOCATION LATITUDE - ',F10.4,2X,'LONGITUDE - ',
     1       F10.4,2X,'RADIUS - ',F9.1)
 2010 FORMAT(52A2,2A2,1X,I3,1X,4A2)
 2011 FORMAT(52A2,2A2,1X,I3,3(1X,I8))
 2020 FORMAT(1X,I3,')',3X,52A2,2A2,1X,I3,1X,4A2)
 2021 FORMAT(1X,I3,')',3X,52A2,2A2,1X,I3,3(1X,I8))
 2022 FORMAT('                                                    ',
     1       '     CTA     POINTS   SDEV      TREND   ELTANG    PROB',
     1       '   NMWLS   TCA            QTIME       LOSTIM ',
     1       ' BIAS              ELT ID     ')
  999 CONTINUE
      RETURN
      END
```

```
C
C     if the third output is desired then output its headers
C
      IF (THIRD) THEN
        IF (IST .EQ. 1) GO TO 708
        IST=1
        WRITE(UNIT,1011)
        WRITE(UNIT,1012)
        WRITE(UNIT,1013)
 708    CONTINUE
        REWIND(23)
C
C       initialize the buffer
C
        I1=1
        CALL FILBS(40B,240,MSG1,I1)
C
C       read in record from third file
C
  709   CONTINUE
C       read a record from the third output file
        READ(23,1014,IOSTAT=ISTAT,ERR=1995,END=1111)(MSG1(I1),I1=1,50)
        IF (ISTAT .NE. 0) THEN
 1995    CONTINUE
         WRITE(1,710)ISTAT
 710     FORMAT(' READ ERROR ON THIRD FILE, ISTAT IS ',I4)
         GO TO 999
        ENDIF
C       write a record onto the output logical unit
        WRITE(UNIT,1021)NLINE,(MSG1(I1),I1=1,50)
        LINE=LINE+1
        NLINE=NLINE+1
C
C       if end of page then skip to the next page
C
        IF (LINE .GE. PAGE) THEN
          RSPGE=PAGE-LINE
          DO 777 I1=1,RSPGE+8
           WRITE(UNIT,1)
 777       CONTINUE
          WRITE(UNIT,1011)
          WRITE(UNIT,1012)
          WRITE(UNIT,1013)
          LINE=3
C         NLINE=1
        ENDIF
        GO TO 709
      ENDIF
 1111 CONTINUE
      IF (FOURTH) THEN
        RSPGE=PAGE-LINE
        DO 888 I1=1,RSPGE+8
          WRITE(UNIT,1)
 888    CONTINUE
        LINE=3
        NLINE=1
      ENDIF
C
C     if the fourth output is desired then display its headers
C
      IF (FOURTH) THEN
        IF (FST .EQ. 1) GO TO 713
         FST=1
         WRITE(UNIT,1015)
         WRITE(UNIT,1016)
         WRITE(UNIT,1017)
  713   CONTINUE
        REWIND(24)
C
C       initialize buffer with blanks
C
        I1=1
        CALL FILBS(40B,240,MSG1,I1)
C
C       read in record from fourth file
C
  714   CONTINUE
C       read a record from the fourth output file
        READ(24,1018,IOSTAT=ISTAT,ERR=1997,END=716)(MSG1(I1),I1=1,50)
        IF (ISTAT .NE. 0) THEN
 1997     CONTINUE
          WRITE(1,715)ISTAT
  715     FORMAT(' READ ERROR ON FOURTH FILE, ISTAT IS ',I4)
          GO TO 999
        ENDIF
C       write a record from the output file
        WRITE(UNIT,1021)NLINE,(MSG1(I1),I1=1,50)
        LINE=LINE+1
        NLINE=NLINE+1
```

```
C
C       initialize buffer with blanks
C
        I1=1
        CALL FILBS(40B,240,MSG1,I1)
C
C       read in record from second file
C
        SPOT = SPOT + 1
 705    CONTINUE
C       read a record from the second display file
        IF ((BENT) .OR. (REALT) .OR. (COS4)) THEN
         IF (ICONTR(SPOT) .EQ. 2HCA) THEN
          READ(22,2010,IOSTAT=ISTAT,ERR=1994,END=707)
     1     (MSG1(I1),I1=1,52),CONTRY(1),CONTRY(2),IUSER,
     1     (TYPE(I1),I1=1,4)
         ELSE
          READ(22,2011,IOSTAT=ISTAT,ERR=1994,END=707)
     1     (MSG1(I1),I1=1,52),CONTRY(1),CONTRY(2),IUSER,
     1     ELTI2,ELTI3,ELTI4
         ENDIF
        ELSE
         READ(22,1010,IOSTAT=ISTAT,ERR=1994,END=707)(MSG1(I1),I1=1,52),
     1    ELTI1,ELTI2,ELTI3,ELTI4
        ENDIF
        IF (ISTAT .NE. 0) THEN
 1994     CONTINUE
          WRITE(1,706)ISTAT
 706      FORMAT(' READ ERROR ON SECOND FILE, ISTAT IS ',I4)
          GO TO 999
        ENDIF
C       write a record onto the output logical unit
        IF ((BENT) .OR. (REALT) .OR. (COS4)) THEN
          IF (ICONTR(SPOT) .EQ. 2HCA) THEN
           WRITE(UNIT,2020)NLINE,(MSG1(I1),I1=1,52),CONTRY(1),
     1     CONTRY(2),IUSER,(TYPE(I1),I1=1,4)
          ELSE
           WRITE(UNIT,2021)NLINE,(MSG1(I1),I1=1,52),CONTRY(1),
     1     CONTRY(2),IUSER,ELTI2,ELTI3,ELTI4
          ENDIF
        ELSE
         WRITE(UNIT,1020)NLINE,(MSG1(I1),I1=1,52),ELTI1,ELTI2,ELTI3,
     1  ELTI4
        ENDIF
        LINE=LINE+1
        NLINE=NLINE+1
C
C       if end of page, skip to next page
C
        IF (LINE .GE. PAGE) THEN
          RSPCE=PAGE -LINE
          DO 2555 I1=1,RSPCE+8
            WRITE(UNIT,1)
 2555     CONTINUE
          WRITE(UNIT,1007)
          WRITE(UNIT,1008)
          IF ((BENT) .OR.(REALT) .OR.(COS4)) THEN
           WRITE(UNIT,2022)
          ELSE
           WRITE(UNIT,1009)
          ENDIF
          LINE=3
C         NLINE=1
        ENDIF
C
C       SEE IF END OF QUERY
C
        IF (NLINE .EQ. POS) THEN
 24      CONTINUE
         CPOS = POS
         READ(14,FMT=1022,END=25)POS,FLAT,FLONG,FRADI
         IF (POS .EQ. CPOS) GO TO 24
         SPOT = SPOT+1
        ENDIF
        GO TO 705
 25     CONTINUE
        POS = 999
        GO TO 705
       ENDIF
 707   CONTINUE
       IF (THIRD ) THEN
         RSPCE=PAGE-LINE
         DO 666 I1=I,RSPCE+8
          WRITE(UNIT,1)
 666     CONTINUE
         NLINE=1
         LINE=3
       ENDIF
```

```
C
C       calculate delta lat, delta long
C
        LATC=LUTLA$P
        LATT=CLAT$P
        LONGT=CLONG$P
        LONGC=LUTLG$P
C       if a specific location was specified then
        IF (CRADI .NE. 0.0) THEN
           DIFFR=DSTNC(CLONG,CLAT,LUTLG,LUTLA)
           DLAT=R$(LATC-LATT)
           DLONG=R$(LONGC-LONGT)$COS(LATC)
C       no specific location was specified
        ELSE
           DIFFR=0
           DLAT=0
           DLONG=0
        ENDIF
        WRITE(UNIT,1019)NLINE,(MSG1(I1),I1=1,48),DIFFR,DLAT,DLONG
        NLINE=NLINE+1
        LINE=LINE+1
 1555   CONTINUE
C
C       if end of page, skip to next page
C
        IF (LINE .GE. PAGE) THEN
          RSPGE=PAGE-LINE
          DO 222 I1=1,RSPGE+8
            WRITE(UNIT,1)
 222      CONTINUE
          WRITE(UNIT,1000)
          WRITE(UNIT,1001)
          WRITE(UNIT,1002)
          LINE=3
C         NLINE=1
        ENDIF
 1225   CONTINUE
C
C       if have read all the records for this query then display
C       the next query location
C
        IF (NLINE .EQ. POS) THEN
            WRITE(UNIT,1)
            WRITE(UNIT,1023)FLAT,FLONG,FRADI
            WRITE(UNIT,1)
            LINE=LINE+3
            CRADI=FRADI
            CLAT=FLAT
            CLONG=FLONG
            READ(14,FMT=1022,IOSTAT=ISTAT,END=1123)POS,FLAT,FLONG,FRADI
            IF (ISTAT .NE. 0) THEN
             WRITE(1,720)ISTAT
             GO TO 999
            ENDIF
            IF (LINE .GE. PAGE) GO TO 1555
            GO TO 1225
        ENDIF
C
         GO TO 701
 1123 POS=999
      GO TO 701
      ENDIF
 703  CONTINUE
      IF (SECOND) THEN
        RSPGE=PAGE-LINE
        DO 444 I1=1,RSPGE+8
          WRITE(UNIT,1)
 444    CONTINUE
        LINE=3
        NLINE=1
      ENDIF
C
C     if second output is desired then display its headers
C
      IF (SECOND) THEN
       IF (SST .EQ. 1) GO TO 704
        SST=1
        WRITE(UNIT,1007)
        WRITE(UNIT,1008)
        IF ((BENT) .OR. (REALT) .OR. (COS4)) THEN
          WRITE(UNIT,2022)
        ELSE
          WRITE(UNIT,1009)
        ENDIF
 704    CONTINUE
        REWIND(22)
        REWIND(14)
        READ(14,FMT=1022)POS,FLAT,FLONG,FRADI
        CPOS = POS
        READ(14,FMT=1022)POS,FLAT,FLONG,FRADI
        SPOT = 0
```

```
FTN4
$FILES(1,1)
      SUBROUTINE SECDY(OTHDAT,SPECON)
     -          SEF-40-00  V00  (840727.1241)
      IMPLICIT NONE
C
C-----------------------------------------------------------------------*
C                                                                       *
C    SARSAT EVALUATION FACILITY                                         *
C                                                                       *
C-----------------------------------------------------------------------*
C                                                                       *
C  DATE      VERSION      DESCRIPTION            AUTHOR                 *
C                                                                       *
C 83/10/05     00           -----               SUZANNE Y. SLINN       *
C                                                                       *
C-----------------------------------------------------------------------*
C                                                                       *
C DESCRIPTION:                                                          *
C     This subroutine displays the second of the four output files.     *
C                                                                       *
C CALLING SEQUENCE:                                                     *
C                                                                       *
C    CALLED FROM: ALOCS                                                 *
C                                                                       *
C        CALL SECDY                                                     *
C                                                                       *
C-----------------------------------------------------------------------*
C
C-----------------------------------------------------------------------*
C                                                                       *
C                        DATA DECLARATIONS                              *
C                                                                       *
C-----------------------------------------------------------------------*
C
C
      COMMON /LOGG/ LODAT(3), HIDAT(3), STDAT, ENDAT,OUTLU(5),RANGE
      INTEGER LODAT, HIDAT, OUTLU, RANGE,TRY(10)
      DOUBLE PRECISION STDAT, ENDAT
C
      REAL LOCLAT, LOCLNG, RADIUS,LUTLA,LUTLG,BIAS
C
C         LOCLAT - latitude of specified location
C         LOCLNG - longitude of specified location
C         RADIUS - maximum distance between specified location and
C                  file locations
C
      COMMON /LOCN/ LOCLAT, LOCLNG, RADIUS
C
      INTEGER IBASE(5), ILEVL(3), IMODE, ISTAT(10), LIST
      INTEGER ITDTE(3), ITTST(3), ITSAR(3), ITMCC(3), ITSPD(3)
      INTEGER IDLUT(3), IDTST(3), IDSAR(3), IDMCC(3)
C
      INTEGER SBUF(68), LBUF(98), MBUF(76), FBUF(50)
C
      COMMON /BASE/ IBASE, ILEVL, IMODE, ISTAT, LIST, ITDTE,
     -              ITTST, ITSAR, ITMCC, ITSPD, IDLUT, IDMCC,
     -              IDSAR, IDTST, SBUF, LBUF, MBUF, FBUF
C
      DOUBLE PRECISION QTIME, SDT
C
C         SDT    - start date
C
C
C         LODAT  - ascii start date YYMMDD
C         STDAT  - low date in seconds from 1980
C         ENDAT  - high date in seconds from 1980
C
C         OUTLU  - output device, LU or filename
C
      INTEGER RBUF(40), IPARM(5), PBUF(33)
      INTEGER LU, RET
      INTEGER SYS, ISYS, LOGLU
C
C         RBUF   - receiving buffer
C         IPARM  - return buffer for RMPAR call
C         PBUF   - return buffer from PARSE
C
      REAL ONEDAY
      INTEGER DATCH(3), LOOP, DUM(7), I1
      INTEGER UNIT
C
C
      LOGICAL LUT, MCC, SAR, TEST, ALL
C
C         LUT    - flag to indicate LUT location detail file selected
C         MCC    - flag to indicate MCC location detail file selected
C         SAR    - flag to indicate SARSTAT incident detail file selected
C         TEST   - flag to indicate FIELD test master file selectedlected
C         ALL    - all files will be searched
C
```

```
      INTEGER KEYS(8,8), PROM(8), RK
      INTEGER FLG, AA, IFBRK,BGBUF(200)
C
C       FLG   - flag to indicate if /E used
C       AA    - dummy parameter for IFBRK
C
C
      COMMON/SELCT/ALLUT,ALSAT,SAT,SX,CX,SPESAT,PRIN,SECOND,THIRD,
     1       FOURTH,PST,SST,TST,FST,UNIT,DIF,BENT,REALT,COS4,CBC12,
     1       CBC24,ALLFR,POSN,START,EXIT,TAPE,SPECIF,SEVENT,PASS,
     1       IMAGE,CONTRY,IUSER,TYPE,ELTANG
      INTEGER CONTRY(3),IUSER,TYPE(4)
      REAL ELTANG
      LOGICAL ALLUT,ALSAT,SAT,SX,CX,PRIN,SECOND,THIRD,FOURTH
      LOGICAL BENT,REALT,COS4,CBC12,CBC24,ALLFR,EXIT,TAPE,SPECIF
      INTEGER SPESAT(5),I,PST,SST,TST,FST,MSG1(80),POSN,START
      INTEGER SEVENT,PASS(4),SPECON
      LOGICAL IMAGE,OTHDAT
      REAL DIF
C
C       ALLUT -flag to indicate if all locations wanted
C       ALSAT -flag to indicate if all satellites wanted
C
C       SAT   -flag to indicate if specific satellites wanted
C       SX    -flag to indicate if all SARSAT satellites wanted
C       CX    -flag to indicate if all COSPAS satellites wanted
C       SPESAT-array that contains the specific satellites desired
C
      EQUIVALENCE(LUTLO,LBUF(28))
      EQUIVALENCE(LUTLC,LBUF(30))
      EQUIVALENCE(BIAS,LBUF(38))
      EQUIVALENCE(TRY(1),MSG1(49))
C       KEYS    - array containing contents of soft keys
C       PROM    - array of values to return for soft keys
C       RK      - actual value returned
C
C*****************************************************************************
C
C                EXECUTABLE  CODE
C
C*****************************************************************************
C
      CALL LGBUF(BGBUF,200)
      ISYS = LOGLU(SYS)
C
C       initialize variables
C
  100 CONTINUE
C
C     initialize buffer with blanks
C
      I1=1
      CALL FILBS(40B,160,MSG1,I1)
C
C     cross track angle in degrees (CTA)
C
      I1=2
      CALL FPNBS(LBUF(34),177744B,MSG1,I1)
C
C     number of frequency measurements  (NMFREQ)
C
      I1=I1+1
      CALL FORM(10,LBUF(42),MSG1,I1,6)
C
C     standard deviation of residuals in Hz (SDEV)
C
      I1=I1+1
      CALL FPNBS(LBUF(74),177744B,MSG1,I1)
C
C     trend factor of residuals in Hz (TREND)
C
      I1=I1+1
      CALL FPNBS(LBUF(76),177744B,MSG1,I1)
C
C     quality factor of CBC data (sun angs) (QUAL)
C
C
C     IF 406 DATA, OUTPUT THE ELT ANGLE INSTEAD
C
      IF ((BENT) .OR.(REALT) .OR. (COS4)) THEN
        I1=I1+1
        CALL FPNBS(ELTANG,177744B,MSG1,I1)
      ELSE
        I1=I1+1
        CALL FORM(10,LBUF(98),MSG1,I1,6)
      ENDIF
C
C     probability of tree solution, as % (PROB)
C
      I1=I1+1
      CALL FORM(10,LBUF(40),MSG1,I1,6)
```

```
C
C     number of WLS iterations (NMWLS)
C
      I1=I1+1
      CALL FORM(10,LBUF(43),MSG1,I1,6)
C
C     time of closest approach in hours from the date of AOS  (TCA)
C
      I1=I1+1
      CALL FPNBS(LBUF(36),177744B,MSG1,I1)
C
C     time of AOS (QTIME)
C
      I1=I1+4
      CALL FPNBS(LBUF(8),177744B,MSG1,I1)
C
C     date of LOS, in hours from date of AOS  (LOSTIM)
C
      I1=I1+4
      CALL FPNBS(LBUF(19),177744B,MSG1,I1)
C
C     ELT frequency bias,expressed Hz (BIAS)
C
      ENCODE(20,123,TRY)BIAS
  123 FORMAT(F7.0)
C
C     Elt id  is written to the file from LBUF(21),LBUF(22),LBUF(23),LBUF(24)
C
C
C     IF 406 DATA, CHANGE THE ELTID CODE OUTPUT
C
      IF ((BENT) .OR. (REALT) .OR. (COS4)) THEN
C
C       IF COUNTRY IS CANADA, OUTPUT BEACON NAME
C
        IF((CONTRY(1) .EQ. 2HCA) .AND. (SPECON .EQ. 1HY))THEN
          IF (OTHDAT) THEN
            WRITE(22,1004)(MSG1(I1),I1=1,52),CONTRY(1),CONTRY(2),
     1        IUSER,TYPE(1),TYPE(2),TYPE(3),TYPE(4)
          ELSE
            WRITE(11,1004)(MSG1(I1),I1=1,52),CONTRY(1),CONTRY(2),
     1        IUSER,TYPE(1),TYPE(2),TYPE(3),TYPE(4)
          ENDIF
        ELSE
          IF (OTHDAT) THEN
            WRITE(22,1005)(MSG1(I1),I1=1,52),CONTRY(1),CONTRY(2),
     1        IUSER,LBUF(22),LBUF(23),LBUF(24)
          ELSE
            WRITE(11,1005)(MSG1(I1),I1=1,52),CONTRY(1),CONTRY(2),
     1        IUSER,LBUF(22),LBUF(23),LBUF(24)
          ENDIF
        ENDIF
       ELSE
       IF (OTHDAT) THEN
        WRITE(22,1006)(MSG1(I1),I1=1,52),LBUF(21),LBUF(22),LBUF(23),
     1    LBUF(24)
       ELSE
        WRITE(11,1006)(MSG1(I1),I1=1,52),LBUF(21),LBUF(22),LBUF(23),
     1      LBUF(24)
       ENDIF
       ENDIF
 1004 FORMAT(52A2,2A2,1X,I3,1X,4A2)
 1005 FORMAT(52A2,2A2,1X,I3,3(1X,I8))
 1006 FORMAT(52A2,4(1X,I8))
      RETURN
      END
```

```
FTN4
@FILES(1,1)
      SUBROUTINE TERTI(OTHDAT)
     -          SEF-40-00  V00  (840726.0904)
      IMPLICIT NONE
C
C---------------------------------------------------------------------$
C                                                                      $
C    SARSAT EVALUATION FACILITY                                        $
C                                                                      $
C---------------------------------------------------------------------$
C                                                                      $
C   DATE        VERSION         DESCRIPTION          AUTHOR            $
C                                                                      $
C  83/10/05       00              ------            SUZANNE Y. SLINN   $
C                                                                      $
C---------------------------------------------------------------------$
C                                                                      $
C DESCRIPTION:                                                         $
C     This subroutine displays the third of the four output files.     $
C                                                                      $
C                                                                      $
C CALLING SEQUENCE:                                                    $
C                                                                      $
C     CALLED FROM: ALOCS                                               $
C                                                                      $
C         CALL TERTI                                                   $
C                                                                      $
C---------------------------------------------------------------------$
C
C
C---------------------------------------------------------------------$
C                                                                      $
C                        DATA DECLARATIONS                             $
C                                                                      $
C---------------------------------------------------------------------$
C
C
C
      COMMON /LOGG/ LODAT(3), HIDAT(3), STDAT, ENDAT,OUTLU(5),RANGE
      INTEGER LODAT, HIDAT, OUTLU, RANGE,TRY(10)
      DOUBLE PRECISION STDAT, ENDAT
C
      REAL LOCLAT, LOCLNG, RADIUS,LUTLA,LUTLG,BIAS
C
C         LOCLAT - latitude of specified location
C         LOCLNG - longitude of specified location
C         RADIUS - maximum distance between specified location and
C                  file locations
C
      COMMON /LOCN/ LOCLAT, LOCLNG, RADIUS
C
      INTEGER IBASE(5), ILEVL(3), IMODE, ISTAT(10), LIST
      INTEGER ITDTE(3), ITTST(3), ITSAR(3), ITMCC(3), ITSPD(3)
      INTEGER IDLUT(3), IDTST(3), IDSAR(3), IDMCC(3)
C
      INTEGER SBUF(68), LBUF(98), MBUF(76), FBUF(50)
C
      COMMON /BASE/ IBASE, ILEVL, IMODE, ISTAT, LIST, ITDTE,
     -              ITTST, ITSAR, ITMCC, ITSPD, IDLUT, IDMCC,
     -              IDSAR, IDTST, SBUF, LBUF, MBUF, FBUF
C
      DOUBLE PRECISION QTIME, SDT
C
C         SDT    - start date
C
C
C         LODAT  - ascii start date YYMMDD
C         STDAT  - low date in seconds from 1980
C         ENDAT  - high date in seconds from 1980
C
C         OUTLU  - output device, LU or filename
C
      INTEGER RBUF(40), IPARM(5), PBUF(33)
      INTEGER LU, RET
      INTEGER SYS, ISYS, LOGLU
C
C         RBUF   - receiving buffer
C         IPARM  - return buffer for RMPAR call
C         PBUF   - return buffer from PARSE
C
      REAL ONEDAY
      INTEGER DATCH(3), LOOP, DUM(7), I1
      INTEGER UNIT
C
C
      LOGICAL LUT, MCC, SAR, TEST, ALL
C
C         LUT    - flag to indicate LUT location detail file selected
C         MCC    - flag to indicate MCC location detail file selected
C         SAR    - flag to indicate SARSTAT incident detail file selected
C         TEST   - flag to indicate FIL9 test master file selectedlected
C         ALL    - all files will be searched
C
```

```
      INTEGER KEYS(8,8), PROM(8), RK
      INTEGER FLG, AA, IFBRK
C
C       FLG   - flag to indicate if /E used
C       AA    - dummy parameter for IFBRK
C
C
      COMMON/SELCT/ALLUT,ALSAT,SAT,SX,CX,SPESAT,PRIM,SECOND,THIRD,
     1       FOURTH,PST,SST,TST,FST,UNIT,DIF,BENT,REALT,COS4,CBC12,
     1       CBC24,ALLFR,POSN,START,EXIT,TAPE,SPECIF,SEVENT,PASS,
     1       IMAGE, CONTRY,IUSER,TYPE,ELTANG
      INTEGER CONTRY(3),IUSER,TYPE(4)
      REAL ELTANG
      LOGICAL ALLUT,ALSAT,SAT,SX,CX,PRIM,SECOND,THIRD,FOURTH
      LOGICAL BENT,REALT,COS4,CBC12,CBC24,ALLFR,EXIT,TAPE,SPECIF
      INTEGER SPESAT(5),1,PST,SST,TST,FST,MSG1(80),POSN,START
      INTEGER SEVENT,PASS(4)
      LOGICAL IMAGE,OTHDAT
      REAL DIF
C
C       ALLUT -flag to indicate if all locations wanted
C       ALSAT -flag to indicate if all satellites wanted
C
C       SAT   -flag to indicate if specific satellites wanted
C       SX    -flag to indicate if all SARSAT satellites wanted
C       CX    -flag to indicate if all COSPAS satellites wanted
C       SPESAT-array that contains the specific satellites desired
C
      EQUIVALENCE(LUTLA,LBUF(20))
      EQUIVALENCE(LUTLG,LBUF(30))
      EQUIVALENCE(BIAS,LBUF(70))
      EQUIVALENCE(TRY,MSG1(35))
C       KEYS    - array containing contents of soft keys
C       PROM    - array of values to return for soft keys
C       RK      - actual value returned
C
C*************************************************************************
C
C                 EXECUTABLE  CODE
C
C*************************************************************************
C
      ISYS = LOGLU(SYS)
C
C       initialize variables
C
  100 CONTINUE
      I1=1
      CALL FILBS(40B,160,MSG1,I1)
C
C     drift
C
      I1=2
      CALL FPNBS(LBUF(40),177744B,MSG1,I1)
C
C     initial estimate of CTA in degrees  (CTAI)
C
      I1=I1+1
      CALL FPNBS(LBUF(66),177744B,MSG1,I1)
C
C     initial estimate of TCA in seconds (TCAI)
C
      I1=I1+1
      CALL FPNBS(LBUF(68),177744B,MSG1,I1)
C
C     initial estimate of BIAS in Hz (BIASI)
C
      ENCODE(20,123,TRY)BIAS
  123 FORMAT(F7.0)
C
C     major axis of error ellipse in Km. (MAJAX)
C
      I1=I1+1
      CALL FPNBS(LBUF(55),177743B,MSG1,I1)
C
C     minor axis of error ellipse in KM. (MINAX)
C
      I1=I1+1
      CALL FPNBS(LBUF(57),177743B,MSG1,I1)
C
C     average of data residuals in Hz (AMEAN)
C
      I1=I1+5
      CALL FPNBS(LBUF(72),177744B,MSG1,I1)
C
      IF (OTHDAT) THEN
        WRITE(23,1004)(MSG1(I1),I1=1,50)
      ELSE
        WRITE(12,1004)(MSG1(I1),I1=1,50)
      ENDIF
 1004 FORMAT(50A2)
      RETURN
      END
```

```
FTN4
$FILES(2,2)
      SUBROUTINE QUADY(OTHDAT)
     -          SEF-40-80  V80  (840726.0905)
      IMPLICIT NONE
C
C-----------------------------------------------------------------------$
C                                                                       $
C     SARSAT EVALUATION FACILTIY                                        $
C                                                                       $
C-----------------------------------------------------------------------$
C                                                                       $
C    DATE      VERSION         DESCRIPTION          AUTHOR              $
C  83/10/05      80              -------           SUZANNE Y. SLINN    $
C                                                                       $
C-----------------------------------------------------------------------$
C                                                                       $
C DESCRIPTION:                                                          $
C     This subroutine displays the fourth of the four output files.     $
C                                                                       $
C CALLING SEQUENCE:                                                     $
C                                                                       $
C    CALLED FROM: ALOCS                                                 $
C                                                                       $
C        CALL QUADY                                                     $
C                                                                       $
C-----------------------------------------------------------------------$
C
C
C-----------------------------------------------------------------------$
C                                                                       $
C                         DATA DECLARATIONS                             $
C                                                                       $
C-----------------------------------------------------------------------$
C
C
C
      COMMON /LOGG/ LODAT(3), HIDAT(3), STDAT, ENDAT,OUTLU(5),RANGE
      INTEGER LODAT, HIDAT, OUTLU, RANGE,TRY(10)
      DOUBLE PRECISION STDAT, ENDAT
C
      REAL LOCLAT, LOCLNG, RADIUS,LUTLA,LUTLG
C
C        LOCLAT - latitude of specified location
C        LOCLNG - longitude of specified location
C        RADIUS - maximum distance between specified location and
C                 file locations
C
      COMMON /LOCN/ LOCLAT, LOCLNG, RADIUS
C
      INTEGER IBASE(5), ILEVL(3), IMODE, ISTAT(10), LIST
      INTEGER ITDTE(3), ITTST(3), ITSAR(3), ITMCC(3), ITSPD(3)
      INTEGER IDLUT(3), IDTST(3), IDSAR(3), IDMCC(3)
C
      INTEGER SBUF(68), LBUF(98), MBUF(76), FBUF(50)
C
      COMMON /BASE/ IBASE, ILEVL, IMODE, ISTAT, LIST, ITDTE,
     -              ITTST, ITSAR, ITMCC, ITSPD, IDLUT, IDMCC,
     -              IDSAR, IDTST, SBUF, LBUF, MBUF, FBUF
C
      DOUBLE PRECISION QTIME, SDT
C
C        SDT    - start date
C
C
C        LODAT  - ascii start date YYMMDD
C        STDAT  - low date in seconds from 1980
C        ENDAT  - high date in seconds from 1980
C
C        OUTLU  - output device, LU or filename
C
      INTEGER RBUF(40), IPARM(5), PBUF(33)
      INTEGER LU, RET
      INTEGER SYS, ISYS, LOGLU
C
C        RBUF   - receiving buffer
C        IPARM  - return buffer for RMPAR call
C        PBUF   - return buffer from PARSE
C
      REAL ONEDAY
      INTEGER DATCH(3), LOOP, DUM(7), I1
      INTEGER UNIT
C
C
      LOGICAL LUT, MCC, SAR, TEST, ALL
C
C        LUT    - flag to indicate LUT location detail file selected
C        MCC    - flag to indicate MCC location detail file selected
C        SAR    - flag to indicate SARSTAT incident detail file selected
C        TEST   - flag to indicate FIELD test master file selectedlected
C        ALL    - all files will be searched
```

```
      INTEGER KEYS(8,8), PROM(8), RK
      INTEGER FLG, AA, IFBRK
C
C       FLG   - flag to indicate if /E used
C       AA    - dummy parameter for IFBRK
C
C
      COMMON/SELCT/ALLUT,ALSAT,SAT,SX,CX,SPESAT,PRIM,SECOND,THIRD,
     1       FOURTH,PST,SST,TST,FST,UNIT,DIF,BENT,REALT,COS4,CBC12,
     1       CBC24,ALLFR,POSN,START,EXIT,TAPE,SPECIF,SEVENT,PASS,
     1       IMAGE,CONTRY,IUSER,TYPE,ELTANG
      INTEGER CONTRY(3),IUSER,TYPE(4)
      REAL ELTANG
      LOGICAL ALLUT,ALSAT,SAT,SX,CX,PRIM,SECOND,THIRD,FOURTH
      LOGICAL BENT,REALT,COS4,CBC12,CBC24,ALLFR,EXIT,TAPE,SPECIF
      INTEGER SPESAT(5),I,PST,SST,TST,FST,MSG1(80),POSN,START
      INTEGER SEVENT,PASS(4)
      LOGICAL IMAGE,OTHDAT
      REAL DIF
C
C       ALLUT -flag to indicate if all locations wanted
C       ALSAT -flag to indicate if all satellites wanted
C
C       SAT   -flag to indicate if specific satellites wanted
C       SX    -flag to indicate if all SARSAT satellites wanted
C       CX    -flag to indicate if all COSPAS satellites wanted
C       SPESAT-array that contains the specific satellites desired
C
      EQUIVALENCE(LUTLA,LBUF(28))
      EQUIVALENCE(LUTLG,LBUF(30))
C       KEYS    - array containing contents of soft keys
C       PROM    - array of values to return for soft keys
C       RK      - actual value returned
C
C*************************************************************************
C
C               EXECUTABLE  CODE
C
C*************************************************************************
C
      ISYS = LOCLU(SYS)
C
C       initialize variables
C
  100 CONTINUE
      I1=1
      CALL FILBS(40B,160,MSG1,I1)
C
C     standard deviation of CTA in degrees (VARCTA)
C
      I1=2
      CALL FPNBS(LBUF(78),177744B,MSG1,I1)
C
C     standard deviation of TCA in seconds (VARTCA)
C
      I1=I1+1
      CALL FPNBS(LBUF(80),177744B,MSG1,I1)
C
C     standard deviation of BIAS in Hz (VARBIA)
C
      I1=I1+1
      CALL FPNBS(LBUF(82),177744B,MSG1,I1)
C
C     standard deviation of DRIFT in Hz/min (VARDRI)
C
      I1=I1+1
      CALL FPNBS(LBUF(84),177744B,MSG1,I1)
C
C     correlation of CTA with TCA (CORCT)
C
      I1=I1+1
      CALL FPNBS(LBUF(86),177744B,MSG1,I1)
C
C     correlation of CTA with BIAS (CORCB)
C
      I1=I1+1
      CALL FPNBS(LBUF(88),177744B,MSG1,I1)
C
C     correlation of CTA with DRIFT (CORCD)
C
      I1=I1+1
      CALL FPNBS(LBUF(90),177744B,MSG1,I1)
C
C     correlation of TCA with BIAS (CORTB)
C
      I1=I1+1
      CALL FPNBS(LBUF(92),177744B,MSG1,I1)
C
C     correlation of TCA with DRIFT (CORTD)
C
      I1=I1+1
      CALL FPNBS(LBUF(94),177744B,MSG1,I1)
```

```
C
C     correlation of BIAS with DRIFT (CORBD)
C
      I1=I1+1
      CALL FPNBS(LBUF(96),177744B,MSG1,I1)
C
      IF (OTHDAT) THEN
       WRITE(24,1004)(MSG1(I1),I1=1,50)
      ELSE
       WRITE(13,1004)(MSG1(I1),I1=1,50)
      ENDIF
 1004 FORMAT(50A2)
      RETURN
      END
```

```
FTN4
      SUBROUTINE SESET (FTNLU,RET)
     1         , SEF -040-00  V00  (831011 1443)
C
      IMPLICIT NONE
C                                                                    *
C--------------------------------------------------------------------*
C                                                                    *
C     SARSAT EVALUATION FACILITY                                     *
C                                                                    *
C                                                                    *
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -*
C                                                                    *
C                                                                    *
C   DATE    VERSION        DESCRIPTION                               *
C                                                                    *
C 83/10/11    00           - - - - - - -                             *
C                                                                    *
C                                                                    *
C--------------------------------------------------------------------*
C                                                                    *
C                                                                    *
C DESCRIPTION:                                                       *
C                                                                    *
C        This subroutine is called from LOCAT.  It prompts           *
C         the terminal for the responses for start date, end date,   *
C         and output device.                                         *
C                                                                    *
C                                                                    *
C                                                                    *
C CALLING SEQUENCE                                                   *
C                                                                    *
C      CALL SESET(FTNLU,RET)                                         *
C                                                                    *
C   ARGUMENTS:                                                       *
C                                                                    *
C                                                                    *
C          FTNLU - FTN unit number for output file.                  *
C                                                                    *
C          RET   - Return code, -1 -> error                          *
C                                0 -> okay.                          *
C                                                                    *
C--------------------------------------------------------------------*
C
C--------------------------------------------------------------------*
C                                                                    *
C                       DATA DECLARATION                             *
C                                                                    *
C--------------------------------------------------------------------*
C
C
C
      COMMON /LOGG/ LODAT(3), HIDAT(3), STDAT, ENDAT, OUTLU(5),
     -              RANGE
C
      INTEGER FTNLU,RET
C        FTNLU  - Internal FTN4X LU
C        RET    - Return code
      INTEGER LODAT, HIDAT, OUTLU, RANGE
      DOUBLE PRECISION STDAT, ENDAT, GTIME
      DOUBLE PRECISION DPRG
C
C        LODAT  - ascii start  date YYMMDD
C        HIDAT  - ascii end date YYMMDD
C        OUTLU  - output device, LU or filename
C        RANGE  - no. of days to be considered
C        DPRG   - d.p. range
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
```

```
      INTEGER DEV,PTRLU,IPARM(5),IBUF(39)
      INTEGER SYS, ISYS, LOGLU
C
C  DEV     - Terminal LU
C  PTRLU   - Printer LU  Set to 6
C  IPARM   - Buffer for RMPAR
C  IBUF    - Input buffer for terminal input
C
      INTEGER LUERR
C
C  LUERR  - Error return for output file
C
      INTEGER DAYS(12),DBUF(3)
C
C  DAYS   - Data vector for days in each month
C  DBUF   - Buffer to store current date, YYMMDD, numeric
C
      INTEGER LDATE(3),HDATE(3)
C
C  LDATE  - Temporary buffer for low date
C  HDATE  - Temporary buffer for high date
C
      INTEGER RET
C
C  RET    - Return code for LOSET
C
      INTEGER RBUF(33)
C
C  RBUF   - Result buffer for PARSE
C
      INTEGER NO,YE,SLSHE,END
C
C  NO     - 2HNO
C  YE     - 2HYE
C  SLSHE  - 2H/E
C  END    - 2HEN
C
      INTEGER I, I1, IERR
C
C          I, I1 - loop control variables
C
C
      DATA NO,YE,SLSHE,END/2HNO,2HYE,27505B,2HEN/
      DATA DAYS/31,28,31,30,31,30,31,31,30,31,30,31/
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  INITIALIZATION!!
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      ISYS = LOGLU(SYS)
      DEV=1
      RET=0
      PTRLU=6
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C

C  Get current date and time from the system.
C  (numeric)
C
      CALL EXEC(11,RBUF,DBUF(1))
      DBUF(2)=0
      CALL DATE(DBUF(1),DBUF(2),RBUF(5))
      DBUF(1)=DBUF(1)-1900
      DBUF(3)=RBUF(5)
```

```
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  Low date
C  Defaults to current YYMMDD
C
   10 CONTINUE
      WRITE(DEV,9000)
      READ(1,9012)(LDATE(I),I=1,3),(HDATE(I),I=1,3)
      IF (LDATE(1) .EQ. 0) LDATE(1)=DBUF(1)
      IF (LDATE(1) .LT. 80 .OR. LDATE(1) .GT. 99) GO TO 840
C
      IF (LDATE(2) .EQ. 0) LDATE(2)=DBUF(2)
      IF (LDATE(2) .LT. 1 .OR. LDATE(2) .GT. 12) GO TO 850
C
      IF (LDATE(3) .EQ. 0) LDATE(3)=DBUF(3)
      DAYS(2)=28
      IF (((LDATE(1)/4)*4) .EQ. LDATE(1)) DAYS(2)=29
      IF (LDATE(3) .LT. 1 .OR. LDATE(3) .GT. DAYS(LDATE(2))) GO TO 860
C
C  Now set up to convert dates  to appropriate
C  formats
C
      I1 = 1
      DO 42 I = 1,3
   42 CALL NUMBS(LDATE(I),2,LODAT,I1)
      I1 = 1
      STDAT = QTIME(LODAT,I1,3,0,0)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  High date
C  Defaults to current YYMMDD
C
   50 CONTINUE
      IF (HDATE(1) .EQ. 0) HDATE(1)=DBUF(1)
      IF (HDATE(1) .LT. 80 .OR. HDATE(1) .GT. 99) GO TO 880
C
      IF (HDATE(2) .EQ. 0) HDATE(2)=DBUF(2)
      IF (HDATE(2) .LT. 1 .OR. HDATE(2) .GT. 12) GO TO 890
C
      IF (HDATE(3) .EQ. 0) HDATE(3)=DBUF(3)
      DAYS(2)=28
      IF (((HDATE(1)/4)*4) .EQ. HDATE(1)) DAYS(2)=29
      IF (HDATE(3) .LT. 1 .OR. HDATE(3) .GT. DAYS(HDATE(2))) GO TO 815
   80 CONTINUE
C
C  Now set up to convert dates and times to appropriate
C  formats
C
      I1 = 1
      DO 43 I = 1,3
   43 CALL NUMBS(HDATE(I),2,HIDAT,I1)
      I1 = 1
      ENDAT = QTIME(HIDAT,I1,3,0,0)
C
C  Check that high date is beyond low date
C
      IF (ENDAT .LT. STDAT) GO TO 870
      DPRG = ((ENDAT-STDAT)/(3600*24.)) + 1
      RANGE = DPRG
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
  100 CONTINUE
C
C  Request output device
C
  105 CONTINUE
      WRITE(DEV,9001)
      READ(DEV,9002) IBUF
      CALL PARSE(IBUF,78,RBUF)
      IF (RBUF(2) .EQ. SLSHE .OR. RBUF(2) .EQ. END) GO TO 801
      I=1
      CALL NAMR(RBUF,IBUF,78,I)
```

```
      IF (RBUF(1) .EQ. 0) RBUF(1)=DEV
      DO 110 I=1,3
  110 OUTLU(I)=RBUF(I)
      OUTLU(4)=RBUF(5)
      OUTLU(5)=RBUF(6)
      IF (RBUF(1) .LT. 77B) THEN
         CALL CNUMD(RBUF(1),OUTLU(3))
         OUTLU(1)=2H
         OUTLU(2)=2H
         ENDIF
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
  190 CONTINUE
      OPEN(UNIT=FTNLU,FILE=OUTLU,ERR=800,IOSTAT=LUERR)
      GO TO 900
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  ERRORS!
C
C  Output file access error
C
  800 CONTINUE
      WRITE(DEV,9003) OUTLU,LUERR
  801 RET=-1
      GO TO 900
C
C  High date day incorrect
C
  815 CONTINUE
      WRITE(DEV,9010)
      GO TO 10
C
C  Low date year incorrect
C
  840 CONTINUE
      WRITE(DEV,9008)
      GO TO 10
C
C  Low date month incorrect
C
  850 CONTINUE
      WRITE(DEV,9009)
      GO TO 10
C
C  Low date day incorrect
C
  860 CONTINUE
      WRITE(DEV,9010)
      GO TO 10
C
C
C  High date precedes low date
C
  870 CONTINUE
      WRITE(DEV,9011)
      GO TO 10
C
C  High date year incorrect
C
  880 CONTINUE
      WRITE(DEV,9008)
      GO TO 10
C
C  High date month incorrect
C
  890 CONTINUE
      WRITE(DEV,9009)
      GO TO 10
```

```
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  EXIT!!
C
  900 CONTINUE
      RETURN
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  FORMATS!
C
 9000 FORMAT('Enter date for start and end of search: ',
     1        'ie  YYMMDD YYMMDD')
 9001 FORMAT("Specify retrieval output device LU: _")
 9002 FORMAT(39A2)
 9003 FORMAT("$ ERROR $ File ",3A2," ",06," ",06," access error "
    C       ,I6," ")
 9004 FORMAT("$ ERROR $ Answer YES or NO ")
 9006 FORMAT("$ ERROR $ Disaster called incorrectly!!")
 9007 FORMAT(" Enter date for stop of search ")
 9008 FORMAT("$ ERROR $ Year incorrect ")
 9009 FORMAT("$ ERROR $ Month incorrect ")
 9010 FORMAT("$ ERROR $ Day incorrect ")
 9011 FORMAT("$ ERROR $ High date precedes low date")
 9012 FORMAT(3I2,1X,3I2)
      END
```

```
$$This file writes the output files onto the magnetic tape
$$
SV,1,,TH
ST,1G,B
IF,2G,EQ,/E,5
ST,2G,B
IF,3G,EQ,/E,3
ST,3G,B
IF,4G,EQ,/E,1
ST,4G,B
IF,5G,EQ,/E,1
ST,5G,B
CA,1
CA,2
CA,3
CA,4
CA,5
```

```
**********************************************************************
**                                                                  *
**    PROCEDURE :LOCAT                                              *
**                                                                  *
:**                                                                 *
**                16  SEP, 1983                                     *
**                                                                  *
**    PROGRAMMER SUZANNE Y. SLINN                                   *
**                                                                  *
**    VERSION    1.0              (840514.1049)                     *
**                                                                  *
**********************************************************************
**                                                                  *
**    DESCRIPTION                                                   *
**                                                                  *
**        Load sequence for LOCAT                                   *
**                                                                  *
**********************************************************************
**                                                                  *
EC
SZ,29
OP,LB
RE,ZLOCAT
RE,ZLOCBL
**RE,ZALOCS
RE,ZOUTPUT
RE,ZOUTP2
**RE,ZPRIMARY
**RE,ZSECOND
**RE,ZTHIRD
**RE,ZFOURTH
RE,ZFREQUENCY
RE,ZOUTDAT
RE,ZSEMCC
RE,ZSESAR
RE,ZSETST
RE,ZRETLL
RE,ZSESE1
RE,ZDSTNC
RE,ZCLONS
SE
SE,ZDOLIB
SE,ZDBASE
SE,ZLTSB3
SEA,ZSEFLB
SEA,ZBULIB
/E

**********************************************************************
**                                                                  *
**    PROCEDURE ;ALOCS                                              *
**                                                                  *
 **                                                                 *
**                14  MAY, 1984                                     *
**                                                                  *
**    PROGRAMMER SUZANNE Y. SLINN                                   *
**                                                                  *
**    VERSION    1.0              (840515.1348)                     *
**                                                                  *
**********************************************************************
**                                                                  *
**    DESCRIPTION                                                   *
**                                                                  *
**        Load sequence for ALOCS                                   *
**                                                                  *
**********************************************************************
**                                                                  *
EC
SZ,29
OP,LB
RE,ZLOCBL
RE,ZALOCS
RE,ZOUTP2
RE,ZPRIMARY
RE,ZSECOND
RE,ZTHIRD
RE,ZFOURTH
#RE,ZFREQUENCY
#RE,ZOUTDAT
RE,ZSEMCC
RE,ZSESAR
RE,ZSETST
RE,ZRETLL
RE,ZSESE1
RE,ZDSTNC
RE,ZCLONS
SE
SE,ZDOLIB
SE,ZDBASE
SE,ZLTSB3
SEA,ZSEFLB
SEA,ZBULIB
/E
```

```
:LOADR,:LOCAT
 SZ,29
 OP,LB
 RE,ZLOCAT
 LOCAT  10042 21126    SEF-40-00  V00  (840727.1242)
 RE,ZLOCBL
 SELCT  21127 21206
 BASE   21207 21731
 LOCN   21732 21737
 LOGG   21740 21761
 **RE,ZALOCS
 RE,ZOUTPUT
 OUTPT  21762 23017    SEF-40-00  V00  (840726.0909)
 RE,ZOUTP2
 OUTP2  23020 24550    SEF-40-00  V00  (840725.1426)
 **RE,ZPRIMARY
 **RE,ZSECOND
 **RE,ZTHIRD
 **RE,ZFOURTH
 RE,ZFREQUENCY
 FREQ   24551 25511    SEF-40-00  V00  (840726.0911)
 RE,ZOUTDAT
 OUTDT  25512 37140    SEF-40-00  V00  (840727.1242)
 RE,ZSEMCC
 SEMCC  37141 40260    SEF-40-00  V00  (830908.1400)
 RE,ZSESAR
 SESAR  40261 41111    SEF-40-00  V00  (830908.1359)
 RE,ZSETST
 SETST  41112 41627    SEF-40-00  V00  (830908.1401)
 RE,ZRETLL
 RETLL  41630 43142    SEF-40-00  V00  (830526.1756)
 RE,ZSESE1
 SESET  43143 44721    SEF -040-00  V00  (840514.0857)
 RE,ZDSTNC
 DSTNC  44722 45123    SEF-30-00  V00  (830228.1642)
 RE,ZCLOMS
 CLOMS  45124 45702    (830705.1728)
 SE

 NAMR   45703 46177  92068-1X021 REV.2013 750701
 LOGLU  46200 46255  92067-1X297 REV.2013 790228
 IDRPL  46256 46504  92067-16185 REV.2040 800710
 IDRPD  46505 46733  92067-16185 REV.2040 800909
 OPEN   46734 47314  92067-16125 REV.2101 810615
 CLOSE  47315 47531  92067-16125 REV.2140 810616
 IDGET  47532 47614  92067-1X298 REV.2013 790314
 RMPAR  47615 47661  92068-1X025 REV.2101 800919
 MESSG  47662 50215  92067-16261 REV.1903 790420
 PARSE  50216 50235  92067-1X281 REV.2013 770714
 CMUMD  50236 50255  92067-1X284 REV.2013 770621
 ERR0   50256 50263  24998-1X250 REV.2140 810506
 .IIO.  50264 50437  24998-1X343 REV.2140 810422
 IFTTY  50440 50525  92067-1X295 REV.2013 790118
 CAPCK  50526 51052  92067-1X310 REV.2013 790801
 $SMVE  51053 51145  92067-1X483 REV.2013 800129
 $ESTB  51146 51162  92067-1X454 REV.2013 790202
 SESSN  51163 51200  92067-16125 REV.1903 780413
 REIO   51201 51325  92067-1X275 REV.2140 810805
 IFBRK  51326 51353  92067-1X276 REV.2013 800129
 LGBUF  51354 51372  24998-1X362 REV.2140 810505
 .FMGB  51373 51646  24998-1X353 REV.2140 810428
 .IOCL  51647 51750  24998-1X305 REV.2101 800731
 .EXIT  51751 52024  24998-1X320 REV.2101 800731
 .FFRW  52025 52123  24998-1X297 REV.2140 810422 N
 READF  52124 53433  92067-16125 REV.2101 810616
 .FMER  53434 53500  24998-1X352 REV.2140 810515
 .EIO.  53501 53553  24998-1X329 REV.2140 810422
 .FMCN  53554 53635  24998-1X345 REV.2140 810415
 .IOER  53636 54003  24998-1X321 REV.2140 810506
 .FMFP  54004 55247  24998-1X346 REV.2140 810519
 .FMO?  55250 55331  24998-1X351 REV.2140 810415
 .UFMP  55332 55346  24998-1X296 REV.2140 810730
 .FMCV  55347 56545  24998-1X333 REV.2140 810810
 .FMUI  56546 57700  24998-1X349 REV.2140 810416
 .FMUR  57701 57747  24998-1X350 REV.2140 810408
 .FMIO  57750 60171  24998-1X348 REV.2140 810406
 .IOOP  60172 60200  24998-1X300 REV.2101 800805
 .IOCM  60201 60244  24998-1X327 REV.2101 801007
 .FFOP  60245 61366  24998-1X301 REV.2140 810724 N
 CREAT  61367 61751  92067-16125 REV.2140 810629
 .FOP?  61752 62027  24998-1X326 REV.2101 800729
 .FFCL  62030 62361  24998-1X306 REV.2140 810728 N
 LOCF   62362 62662  92067-16125 REV.1903 781110
 RWNDF  62663 62747  92067-16125 REV.1903 780724
 POST   62750 62776  92067-16125 REV.1903 740801
```

```
 .DCO    62777 63047   24998-1X037 REV.2001 790417
 .DDI    63050 63350   24998-1X040 REV.2001 781021
 .IOCN   63351 63357   24998-1X315 REV.2101 800731
 .FFCN   63360 63610   24998-1X316 REV.2140 800813 N
POSNT    63611 64151   92067-16125 REV.1903 790316
 .FIOI   64152 64235   24998-1X322 REV.2140 810730
 .RTOI   64236 64350   24998-1X063 REV.2013 791230
 .DMP    64351 64516   24998-1X045 REV.2001 780818
 .DDE    64517 64530   24998-1X039 REV.2001 780818
 .DIN    64531 64536   24998-1X042 REV.2001 780818
 .FCM    64537 64553   24998-1X182 REV.2001 750701
DCOS     64554 64607   24998-1X094 REV.2001 750701
DSIN     64610 65026   24998-1X144 REV.2001 750701
ENTIX    65027 65204   24998-1X189 REV.2001 750701
XPOLY    65205 65262   24998-1X160 REV.2001 750801
 .ASIN   65263 65366   24998-1X373 REV.2101 800222
 .DMG    65367 65376   24998-1X046 REV.2001 780818
 .DADS   65377 65506   24998-1X036 REV.2001 780818
 .DIS    65507 65536   24998-1X044 REV.2001 780818
OVRD     65537 65537   92067-16125 REV.1903 780526
IDRP     65540 67143   92067-16185 REV.2026 800305
R/W$     67144 67302   92067-16125 REV.2101 801013
$OPEN    67303 67457   92067-16125 REV.1903 790103
RW$UR    67460 70031   92067-16125 REV.2101 800303
RWND$    70032 70161   92067-16125 REV.2101 810617
VSCBA    70162 70227   92067-16261 REV.1903 790202
$PARS    70230 70450   92067-1X282 REV.2013 780811
$CVT3    70451 70536   92067-1X287 REV.2013 770621
LURQ     70537 71151   92067-1X270 REV.2013 791024
NAM      71152 71246   92067-16125 REV.1903 740801
P_PAS    71247 71275   92067-16125 REV.1903 740801
LUTRU    71276 71404   92067-1X308 REV.2013 790223
$ALRM    71405 71522   92067-1X271 REV.2013 770715
 .OWNR   71523 71570   92067-1X324 REV.2013 780919
LIMEM    71571 71611   92067-1X477 REV.2013 790126
PNAME    71612 71662   92068-1X035 REV.2101 800919
 .OPN?   71663 71706   24998-1X325 REV.2101 800803
ERQ.E    71707 71707   24998-1X249 REV.2001 750701
 .FMIN   71710 72167   24998-1X344 REV.2140 810427
PAU.E    72170 72170   24998-1X254 REV.2001 750701
SE.ZDQLIB
DQFND    72171 72317     SEF-36-03  V00  (830329.0951)
DQGET    72320 72533     SEF-36-04  V00  (830329.0951)
DQCLS    72534 72614     SEF-36-10  V00  (830329.0951)
DBBLK    72615 73614     SEF-36-11  V00  (830329.0951)
DQMOV    73615 73654     SEF-36-12  V00  (830329.0951)
SE.ZDBASE
SE.ZLTSB3
DTIMA    73655 73744     SEF-25-00  V00  (830727.1002)
SEA.ZSEFLB
QTIME    73745 74121     SEF-22-00  V00  (830221.1504)
PTIME    74122 74320     SEF-XX-00  V00  (830224.0922)
DTIME    74321 74622     SEF-XX-00  V00  (830224.0922)
DATE     74623 74745     SEF-XX-00  V00  (830224.0922)
FORM     74746 75265   Format integer numbers  V01   (840404.0945)
NUMBR    75266 75440   Decode integer numbers  V01   (840404.0945)
SKIP     75441 75475   Skip over a character string  V00   (840404.0945)
GETCH    75476 75601   Get a character from a string  V01   (840404.0945)
PUTCH    75602 75701   Put a character into a string  V01   (840404.0945)
FPNBS    75702 76300     SEF-XX-00  V00  (830602.1010)
SCHAR    76301 76365     SEF-51-16  V00  (830602.1010)
FUNKY    76366 76752     SEF-51-17  V00  (830602.1010)
SEA.ZBULIB
MOVBS    76753 77002     V01  08-JAN-80  08190-07
NUMBS    77003 77174     V01  08-JAN-80  08190-09
FILBS    77175 77224     V01  08-JAN-80  08190-15
/E

 .RTOR   77225 77317   24998-1X064 REV.2026 800509
 .ITOI   77320 77436   24998-1X055 REV.2013 791017
AINT     77437 77537   24998-1X134 REV.2001 750701
MOD      77540 77567   24998-1X107 REV.2001 751101

29 PAGES RELOCATED    29 PAGES REQ'D    NO PAGES EMA      NO PAGES MSEG
LINKS:BP    PROGRAM:LB   LOAD:TE   COMMON:NC
 /LOADR:LOCAT  READY AT 10:31 AM  MON., 17  SEPT, 1984

 /LOADR:$END
```

```
LOADR,:ALOCS
 SZ,29
 OP,LB
 RE,ZLOCRL
 SELCT  10042 10121
 BASE   10122 10644
 LOCN   10645 10652
 LOGG   10653 10674
 RE,ZALOCS
 ALOCS  10675 22155
 BITSG  22156 22275
 RE,ZOUTP2
 OUTP2  22276 24026    SEF-40-00  V00  (840725.1426)
 RE,ZPRIMARY
 PRIMR  24027 26554    SEF-40-00  V00  (840725.1427)
 RE,ZSECOND
 SECDY  26555 30332    SEF-40-00  V00  (840727.1241)
 RE,ZTHIRD
 TERTI  30333 31150    SEF-40-00  V00  (840726.0904)
 RE,ZFOURTH
 QUADY  31151 32016    SEF-40-00  V00  (840726.0905)
 *RE,ZFREQUENCY
 *RE,ZOUTDAT
 RE,ZSEMCC
 SEMCC  32017 33136    SEF-40-00  V00  (830908.1400)
 RE,ZSESAR
 SESAR  33137 33767    SEF-40-00  V00  (830908.1359)
 RE,ZSETST
 SETST  33770 34505    SEF-40-00  V00  (830908.1401)
 RE,ZRETLL
 RETLL  34506 36020    SEF-40-00  V00  (830526.1756)
 RE,ZSESE1
 SESET  36021 37577    SEF -040-00  V00  (840514.0857)
 RE,ZDSTNC
 DSTNC  37600 40001    SEF-30-00  V00  (830228.1642)
 RE,ZCLONS
 CLONS  40002 40560    (830705.1728)
 SE

 NAMR   40561 41055  92068-1X021 REV.2013 750701
 LOGLU  41056 41133  92067-1X297 REV.2013 790228
 IDRPL  41134 41362  92067-16185 REV.2040 800710
 IDRPD  41363 41611  92067-16185 REV.2040 800909
 OPEN   41612 42172  92067-16125 REV.2101 810615
 CLOSE  42173 42407  92067-16125 REV.2140 810616
 IDGET  42410 42472  92067-1X298 REV.2013 790314
 RMPAR  42473 42537  92068-1X025 REV.2101 800919
 MESSS  42540 43073  92067-16261 REV.1903 790420
 PARSE  43074 43113  92067-1X281 REV.2013 770714
 CNUMD  43114 43133  92067-1X284 REV.2013 770621
 ERRO   43134 43141  24998-1X250 REV.2140 810506
  .IIO  43142 43315  24998-1X343 REV.2140 810422
 IFTTY  43316 43403  92067-1X295 REV.2013 790118
 CAPCK  43404 43730  92067-1X318 REV.2013 790801
 $SMVE  43731 44023  92067-1X483 REV.2013 800129
 $ESIB  44024 44040  92067-1X454 REV.2013 790202
 SESSN  44041 44056  92067-16125 REV.1903 780413
 LGBUF  44057 44075  24998-1X362 REV.2140 810505
  .FMGB 44076 44351  24998-1X353 REV.2140 810428
  .IOCL 44352 44453  24998-1X305 REV.2101 800731
  .EXIT 44454 44527  24998-1X320 REV.2101 800731
  .FFRW 44530 44626  24998-1X297 REV.2140 810422 N
 READF  44627 46136  92067-16125 REV.2101 810616
  .FMER 46137 46203  24998-1X352 REV.2140 810515
  .EIO  46204 46256  24998-1X329 REV.2140 810422
  .FMCN 46257 46340  24998-1X345 REV.2140 810415
  .IOER 46341 46506  24998-1X321 REV.2140 810506
  .FMFP 46507 47752  24998-1X346 REV.2140 810519
  .FMO? 47753 50034  24998-1X351 REV.2140 810415
  .UFMP 50035 50051  24998-1X296 REV.2140 810730
  .FMCV 50052 51250  24998-1X333 REV.2140 810810
  .FMUI 51251 52403  24998-1X349 REV.2140 810416
  .FMUR 52404 52452  24998-1X350 REV.2140 810408
  .FMIO 52453 52674  24998-1X348 REV.2140 810406
  .FPAU 52675 53000  24998-1X324 REV.2101 800731
 PAU.E  53001 53001  24998-1X254 REV.2001 750701
  .IOOP 53002 53010  24998-1X308 REV.2101 800805
  .IOCM 53011 53054  24998-1X327 REV.2101 801007
  .FFOP 53055 54176  24998-1X301 REV.2140 810724 N
 CREAT  54177 54561  92067-16125 REV.2140 810629
  .FOP? 54562 54637  24998-1X326 REV.2101 800729
  .FFCL 54640 55171  24998-1X306 REV.2140 810728 N
 LOCF   55172 55472  92067-16125 REV.1903 781110
 RWNDF  55473 55557  92067-16125 REV.1903 780724
 POST   55560 55606  92067-16125 REV.1903 740801
  .DCO  55607 55657  24998-1X037 REV.2001 790417
  .DDI  55660 56160  24998-1X040 REV.2001 781021
  .FIOI 56161 56244  24998-1X322 REV.2140 810730
  .RIOI 56245 56357  24998-1X063 REV.2013 791230
  .DMP  56360 56525  24998-1X045 REV.2001 780818
  .DDE  56526 56537  24998-1X039 REV.2001 780818
  .DIN  56540 56545  24998-1X042 REV.2001 780818
  .ITOI 56546 56664  24998-1X055 REV.2013 791017
  .FCM  56665 56701  24998-1X102 REV.2001 750701
```

```
.ASIN   56702 57005   24998-1X373 REV.2101 800222
.DNG    57006 57015   24998-1X046 REV.2001 780818
.DADS   57016 57125   24998-1X036 REV.2001 780818
OVRD.   57126 57126   92067-16125 REV.1903 780526
IDRP    57127 60532   92067-16185 REV.2026 800305
R/W$    60533 60671   92067-16125 REV.2101 801013
$OPEN   60672 61046   92067-16125 REV.1903 790103
RW$UB   61047 61420   92067-16125 REV.2101 800303
RWND$   61421 61550   92067-16125 REV.2101 810617
VSCBA   61551 61616   92067-16261 REV.1903 790202
$PARS   61617 62037   92067-1X282 REV.2013 780811
$CVT3   62040 62125   92067-1X287 REV.2013 770621
LURQ    62126 62540   92067-1X270 REV.2013 791024
MAN.    62541 62635   92067-16125 REV.1903 740801
P.PAS   62636 62664   92067-16125 REV.1903 740801
LUTRU   62665 62773   92067-1X308 REV.2013 790223
$ALRN   62774 63111   92067-1X271 REV.2013 770715
.OWNR   63112 63157   92067-1X324 REV.2013 780919
LIMEM   63160 63200   92067-1X477 REV.2013 790126
REIO    63201 63325   92067-1X275 REV.2140 810805
PNAME   63326 63376   92068-1X035 REV.2101 800919
.OPN?   63377 63422   24998-1X325 REV.2101 800803
ERR.E   63423 63423   24998-1X249 REV.2001 750701
.FMIN   63424 63703   24998-1X344 REV.2140 810427
SE,ZDQLIB
DQOPN   63704 63764     SEF-36-01  V00  (830329.0951)
DQFND   63765 64113     SEF-36-03  V00  (830329.0951)
DQGET   64114 64327     SEF-36-04  V00  (830329.0951)
DQINF   64330 64460     SEF-36-08  V00  (830329.0951)
DBBLK   64461 65460     SEF-36-11  V00  (830329.0951)
DQMOV   65461 65520     SEF-36-12  V00  (830329.0951)
SE,ZDBASE
SE,ZLTSB3
DTIMA   65521 65610     SEF-25-00  V00  (830727.1002)
SEA.ZSEFLB
QTIME   65611 65765     SEF-22-00  V00  (830221.1504)
PTIME   65766 66164     SEF-XX-00  V00  (830224.0922)
DTIME   66165 66466     SEF-XX-00  V00  (830224.0922)
DATE    66467 66611     SEF-XX-00  V00  (830224.0922)
FORM    66612 67131   Format integer numbers  V01   (840404.0945)
NUMBR   67132 67304   Decode integer numbers  V01   (840404.0945)
SKIP    67305 67341   Skip over a character string  V00   (840404.0945)
GETCH   67342 67445   Get a character from a string  V01   (840404.0945)
PUTCH   67446 67545   Put a character into a string  V01   (840404.0945)
FPNBS   67546 70144     SEF-XX-00  V00  (830602.1010)
SCHAR   70145 70231     SEF-51-16  V00  (830602.1010)
SEA.ZBULIB
MOVBS   70232 70261     V01  08-JAN-80  08190-07
NUMBS   70262 70453     V01  08-JAN-80  08190-09
FILBS   70454 70503     V01  08-JAN-80  08190-15
/E

 RTOR   70504 70576   24998-1X064 REV.2026 800509
AINT    70577 70677   24998-1X134 REV.2001 750701
MOD     70700 70727   24998-1X107 REV.2001 751101

26 PAGES RELOCATED    29 PAGES REQ'D    NO PAGES EMA      NO PAGES MSEG
LINKS:BP    PROGRAM:LB   LOAD:TE   COMMON:NC
 /LOADR:AL8CS  READY AT 10:36 AM  MON., 17  SEPT, 1984

 /LOADR:$END
```

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATING ACTIVITY
Defence Research Establishment Ottawa
Department of National Defence
Ottawa, Ontario, Canada K1A 0Z4

2a. DOCUMENT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP

3. DOCUMENT TITLE

LOCAT - A DATA RETRIEVAL PROGRAM

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
TECHNICAL NOTE

5. AUTHOR(S) (Last name, first name, middle initial)

SLINN, Suzanne Y.

6. DOCUMENT DATE	7a. TOTAL NO OF PAGES	7b. NO OF REFS
DECEMBER 1984	71	

8a. PROJECT OR GRANT NO	9a. ORIGINATOR'S DOCUMENT NUMBER(S)
	DREO TECHNICAL NOTE NO. 84-30
8b. CONTRACT NO	9b. OTHER DOCUMENT NO.(S) (Any other numbers that may be assigned this document)

10. DISTRIBUTION STATEMENT

UNLIMITED DISTRIBUTION

11. SUPPLEMENTARY NOTES	12. SPONSORING ACTIVITY

13. ABSTRACT

The computer program, LOCAT, a data retrieval package for the SARSAT Evaluation Facility database, is described for use on an HP-1000 computer. The program provides the user with flexibility over what data is to be retrieved from the database. The following input options are available:

- satellite(s)
- frequency(s)
- location and radius
- date range

The data retrieved is presented in the form of four outputs, each of which provides different information.

LOCAT is documented in terms of a brief description of the package, its capabilities, a guide on how to use it, and how to compile and load it. The source code for the routines written is provided in the Appendices.

DSIS

77-0n5

UNCLASSIFIED

Security Classification

KEY WORDS

LUT
SEF
MCC
RCC
DATABASE
FORTRAN
HP-1000 COMPUTER
ELT
EPIRB

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the organization issuing the document.

2a. DOCUMENT SECURITY CLASSIFICATION: Enter the overall security classification of the document including special warning terms whenever applicable.

2b. GROUP: Enter security reclassification group number. The three groups are defined in Appendix 'M' of the DRB Security Regulations.

3. DOCUMENT TITLE: Enter the complete document title in all capital letters. Titles in all cases should be unclassified. If a sufficiently descriptive title cannot be selected without classification, show title classification with the usual one-capital-letter abbreviation in parentheses immediately following the title.

4. DESCRIPTIVE NOTES: Enter the category of document, e.g. technical report, technical note or technical letter. If appropriate, enter the type of document, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the document. Enter last name, first name, middle initial. If military, show rank. The name of the principal author is an absolute minimum requirement.

6. DOCUMENT DATE: Enter the date (month, year) of Establishment approval for publication of the document.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the document.

8a. PROJECT OR GRANT NUMBER: If appropriate, enter the applicable research and development project or grant number under which the document was written.

8b. CONTRACT NUMBER: If appropriate, enter the applicable number under which the document was written.

9a. ORIGINATOR'S DOCUMENT NUMBER(S): Enter the official document number by which the document will be identified and controlled by the originating activity. This number must be unique to this document.

9b. OTHER DOCUMENT NUMBER(S): If the document has been assigned any other document numbers (either by the originator or by the sponsor), also enter this number(s).

10. DISTRIBUTION STATEMENT: Enter any limitations on further dissemination of the document, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this document from their defence documentation center."

(2) "Announcement and dissemination of this document is not authorized without prior approval from originating activity."

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document, even though it may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall end with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (TS), (S), (C), (R), or (U).

The length of the abstract should be limited to 20 single-spaced standard typewritten lines, 7½ inches long.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a document and could be helpful in cataloging the document. Key words should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context.

END

FILMED

8-85

DTIC