

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

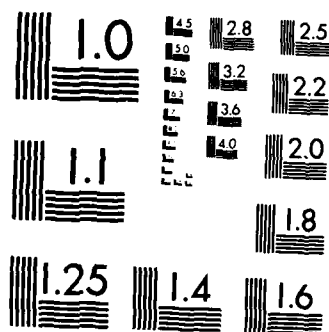
F/G 9/2

NL

END

FILMED

OTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A156 004



National
Defence

Défense
nationale



LOCAT - A DATA RETRIEVAL PROGRAM

by

Suzanne Y. Slinn

DTIC FILE COPY

DTIC
ELECTE
JUN 27 1985
S
A

DEFENCE RESEARCH ESTABLISHMENT OTTAWA
TECHNICAL NOTE 84-30

Canada

This document has been approved
for public release and sale; its
distribution is unlimited.

December 1984
Ottawa

85 6 10 008



National
Defence

Défense
nationale

2

LOCAT - A DATA RETRIEVAL PROGRAM

by

Suzanne Y. Slinn
SARSAT Project Office
Electronics Division

DTIC
2025 RELEASE UNDER E.O. 14176

DEFENCE RESEARCH ESTABLISHMENT OTTAWA
TECHNICAL NOTE 84-30

PCN
33X00

December 1984
Ottawa

For information only - not to be used for reproduction

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 résumé 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'utilisateur 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'utilisateur 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

	<u>Page</u>
1.0 INTRODUCTION	1
2.0 PROGRAM OVERVIEW	1
2.1 DESIGN PHILOSOPHY	2
2.2 SATELLITE AND LOCATION SOFT KEY DISPLAY	3
2.3 FREQUENCY SOFT KEY DISPLAY.	4
2.4 OUTPUT SOFT KEY DISPLAY	5
2.4.1 Primary Output File.	6
2.4.2 Second Output File	7
2.4.3 Third Output File.	8
2.4.4 Fourth Output File	8
3.0 USER'S GUIDE	9
4.0 COMPILING AND LOADING.	10
5.0 SUMMARY COMMENTS	11
APPENDIX A: SAMPLE RUN.	13
APPENDIX B: SOURCE CODE LISTINGS.	17

LIST OF FIGURES

FIGURE 1: LOCAT - Calling Sequence	2
FIGURE 2: SAT/LOC Soft Key Display	3
FIGURE 3: Frequency Soft Key Display	4
FIGURE 4: Output Soft Key Display.	5

AI



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 Centre's (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.

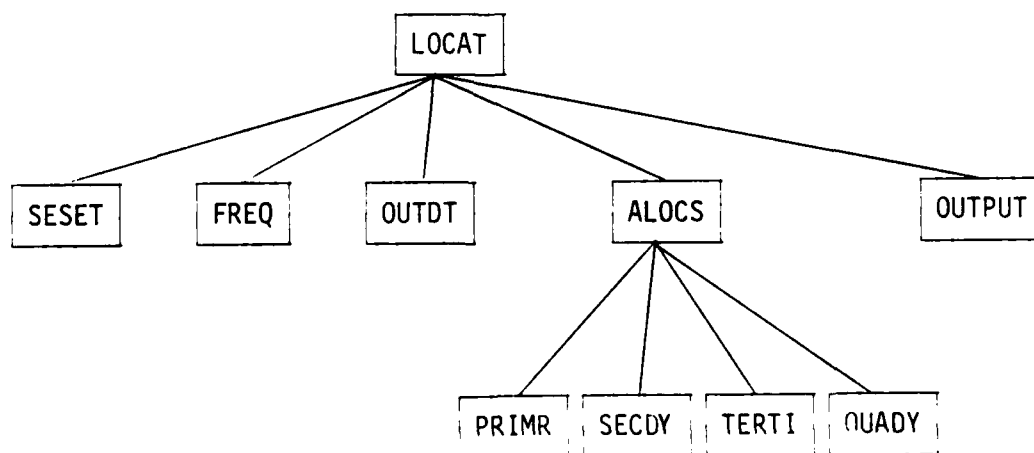


Figure 1: LOCAT - Calling Sequence.

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.

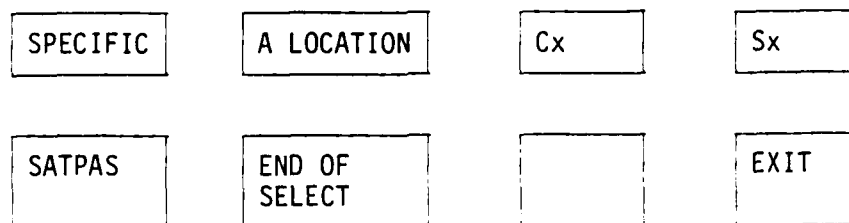


Figure 2: SAT/LOC Soft Key Display

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.

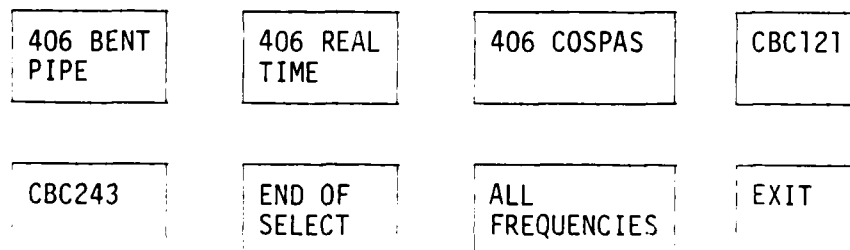


Figure 3: Frequency Soft Key Display

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.

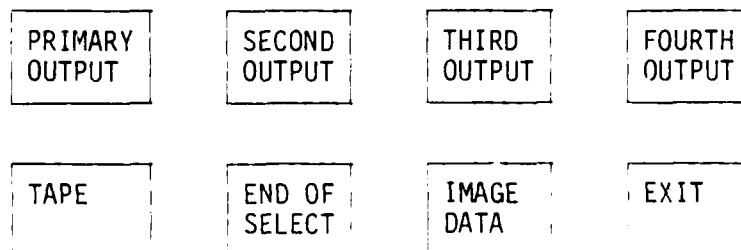


Figure 4: Output Soft Key Display.

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 fourth output file is to be put on tape
      IF FOURTH=1 THEN
        IBUF1(I)=2M/E
        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)=2M/E
      J=-28(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
C 900 CONTINUE
C      close the database
      CALL DDCLS(IRASE,IDLUT,1,ISTAT)
      IF (ISTAT(1) EQ 0) GO TO 999
      WRITE(LU,1005) ISTAT(1)
C
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
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(S(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(SA2)
      END

```

```

CLOSE(30)
C read the LUTELF records and determine if they meet the desired criteria
C CALL ALOCS
C CALL EXEC(9,6HALOCS )
C WRITE(LU,1011)
C READ(LU,1012)ANSWER
C OPEN(30,FILE='SCRATCH',STATUS='OLD')
1023 READ(30,1023)POSN,START
C FORMAT(15,1X,15)
C CLOSE(30)
C IF ((POSN - START) .NE. 0) FOUND = .TRUE.
C IF (ANSWER .EQ. 2HY ) GO TO 5
C IF (FOUND) CALL OUTP2
C IF (FOUND) CALL OUTDT
C store the output files onto tape if desired
C IF (FOUND) THEN
C IF (TAPE) THEN
C I=7
C IBUF(I)=2HHE
C IBUF(I+1)=2HAD
C IBUF(I+2)=2HR,
C I=I+3
C check if the primary output file is to be put on tape
C IF (PRIM) THEN
C IBUF(I)=2HPR
C IBUF(I+1)=2HIM
C IBUF(I+2)=2HAR
C IBUF(I+3)=2HY,
C I=I+4
C ENDIF
C check if the second output file is to be put on tape
C IF (SECOND) THEN
C IBUF(I)=2HSE
C IBUF(I+1)=2HCO
C IBUF(I+2)=2HND
C IBUF(I+3)=2H,
C I=I+4
C ENDIF
C check if the third output file is to be put on tape
C IF (THIRD) THEN
C IBUF(I)=2HTH
C IBUF(I+1)=2HIR
C IBUF(I+2)=2HD,
C I=I+3
C ENDIF
C check if the fourth output file is to be put on tape
C IF (FOURTH) THEN
C IBUF(I)=2HFO
C IBUF(I+1)=2HUR
C IBUF(I+2)=2HTH
C IBUF(I+3)=2H,
C I=I+4
C ENDIF
C IF (IMAGE) THEN
C check if the primary output file is to be put on tape
C IF (PRIM) THEN
C IBUF(I)=2HIP
C IBUF(I+1)=2HRI
C IBUF(I+2)=2HMA
C IBUF(I+3)=2HRY
C IBUF(I+4)=2H,
C I=I+5
C ENDIF
C check if the second output file is to be put on tape
C IF (SECOND) THEN
C IBUF(I)=2HIS
C IBUF(I+1)=2HEC
C IBUF(I+2)=2HOM
C IBUF(I+3)=2HD,
C I=I+4
C ENDIF
C check if the third output file is to be put on tape
C IF (THIRD) THEN
C IBUF(I)=2HIT
C IBUF(I+1)=2HNI
C IBUF(I+2)=2HRD
C IBUF(I+3)=2H,
C I=I+4
C ENDIF

```

```

C      clear the 'R' from the select array
C
C      DO 4 RK=1,8
C        KEYS(1,RK)=2H
C      4 CONTINUE
C
C      210 CONTINUE
C
C      if specific satellites wanted, read the desired satellites
C
C      DO 114 I=1,5
C        SPESAT(I)=2H
C      114
C
C      IF (SAT) THEN
C        WRITE(LU,1008)
C        READ(LU,1009)(SPESAT(I),I=1,5)
C      ENDIF
C
C      read desired lat, long, radius if not all the locations are desired
C
C      IF ( NOT (ALLUT)) THEN
C        WRITE (LU,1000)
C        WRITE (LU,1001)
C        CALL RETLL (LOCLAT, LOCLNG, FLG)
C        IF (FLG .EQ. 1) GO TO 9999
C
C
C      50  RADIUS = 10
C        CALL CLONS(6HPRESP ,LU,SYS,
C          -      12H, .1, I:Radius, -12, IPARM)
C        IF (IPARM(1) .NE. 0) RADIUS = IPARM(1)
C        WRITE(1,554)
C      554 FORMAT(' ENTER THE REGION AND CASE NUMBER OF THIS LOCATION. ')
C        WRITE(1,557)
C      557 FORMAT(' THE CODES FOR THE REGIONS ARE:  HZ - HALIFAX, . / ,
C          1 'TR -TRENTON, ED - EDMONTON, YJ -VICTORIA. ')
C        WRITE(1,556)
C      556 FORMAT(' ENTER THEM ACCORDING TO THIS EXAMPLE:  TR 1111')
C        READ(1,555)REGION,CASENO
C      555 FORMAT(A2,1X,I4)
C      ENDIF
C
C      determine if specific SATPAS is desired
C
C      IF (SPECIF) THEN
C        WRITE(LU,1013)
C        READ(LU,1014)(PASS(I),I=1,4)
C      ENDIF
C
C      determine which of the four outputs are desired
C
C      IF (POSN .EQ. 1) CALL OUTPT
C      IF (POSN .NE. 1) THEN
C        OPEN(14,FILE='HEADR',Iostat=IOS,STATUS='OLD')
C        IF (IOS .NE. 0) WRITE(LU,2018)IOS
C        IF (IOS .NE. 0 ) GO TO 999
C      2018 FORMAT(' ERROR IN OPENING HEADR FILE ',I4)
C        DO 65 I=1,(PLACE-1)
C          READ(14,1017)I
C        1017 READ(14,1017)I
C          FORMAT(A2)
C        65 CONTINUE
C      ENDIF
C
C      IF (EXIT) GO TO 900
C      determine which of the frequencys are desired
C      CALL FREQ
C      IF (EXIT) GO TO 900
C
C      display lat, long, radius
C
C      WRITE(14,1010)POSN,LOCLAT,LOCLNG,RADIUS,REGION,CASENO
C      PLACE=PLACE+1
C      START=POSN
C      CLOSE(14)
C
C      IF (IFBRK(AA) LT 0) GO TO 999
C
C      Write the common variables to a scratch file
C
C      OPEN(30,FILE='SCRATCH',STATUS='OLD')
C      WRITE(30,1010)ALLUT,ALSAT,SAT,SA,CX,(SPESAT(I),I=1,5),PRIN,
C        1 SECOND, THIRD, FOURTH
C      1010 FORMAT(5(I1,1X),SA2,1X,4(I1,1X))
C      WRITE(30,1019)PST,SSST,IST,FST,UNIT,DIF,BENT,REALT,COS4,CRC12,
C        1 CBC24,ALLFR,POSN,START,EXIT,TAPE,SPECIF,SEVENT,
C        1 (PASS(I),I=1,4)
C      1019 FORMAT(5(I5,1X),F10,3,1X,6(I1,1X),2(I5,1X),3(I1,1X),15,1X,4A2)
C      WRITE(30,1020)(LOBAT(I),I=1,3),(HIDAT(I),I=1,3),STDAT,ENDAT,
C        1 (OUTLUT(I),I=1,5),RANGE,LOCLAT,LOCLNG,RADIUS,IMAGE,REGION,
C        1 CASENO,PLACE,(IBASE(I),I=1,5)
C      1020 FORMAT(3A2,1X,3A2,1X,F10,3,1X,F10,3,1X,SA2,1X,10,1X,3(F10,3,1X),
C        1 L1,A2,1X,14,1X,I3,1X,SA2)
C

```

```

C          EXECUTABLE CODE
C *****
C
C      LU=1
C      DO 1 I=1,5
C 1      IBASE(I)=2H
C      WRITE(LU,1015)
C      READ(LU,1016)(IBASE(I),I=1,5)
C      ISYS = LOGLU(SYS)
C      POSN = 1
C      FOUND = .FALSE.
C      EXIT = .FALSE.
C      TAPE = .FALSE.
C      REGION = 2H
C      CASENO = 0
C      PLACE=1
C
C      initialize variables
C
C 5      CONTINUE
C
C      ALLUT= TRUE.
C      ALSAT= TRUE.
C      SAT= FALSE.
C      CX= FALSE
C      SX= FALSE.
C      SPECIF= .FALSE.
C
C      UNIT = 99
C      LUT = .FALSE.
C      MCC = .FALSE.
C      SAR = .FALSE.
C      TEST= .FALSE.
C      ALL = .FALSE.
C      LOCLAT = 0.0
C      LOCLNG = 0.0
C      RADIUS = 0.0
C
C      get other input variables
C
C      CALL SESET(UNIT,RET)
C      IF (RET .NE. 0) GO TO 999
C
C      search files specified
C
C 100 CONTINUE
C      CALL FUNKY (KEYS,PRON)
C      WRITE (LU,1003)
C 101 CALL REID(1,LU,RK,-2)
C      RK = RK -2H0
C      IF (RK .LT. 1 .OR. RK .GT. 8) GO TO 101
C      IF (RK .EQ. 8) THEN
C          EXIT= TRUE.
C          GO TO 900
C      ENDIF
C      IF (RK .EQ. 7) GO TO 101
C
C      set 8 in key selected and check for further selects
C
C      IF (KEYS(1,RK) .EQ. 2H ) THEN
C          KEYS(1,RK) = 2H8
C          IF (RK .EQ. 1) SAT = .TRUE.
C          IF (RK .EQ. 2) ALLUT = .FALSE.
C          IF (RK .EQ. 3) CX = .TRUE.
C          IF (RK .EQ. 4) SX = .TRUE.
C          IF (RK .EQ. 5) SPECIF = .TRUE.
C          GO TO 102
C      ENDIF
C
C      IF (KEYS(1,RK) .EQ. 2H8 ) THEN
C          KEYS(1,RK) = 2H
C          IF (RK .EQ. 1) SAT = .FALSE.
C          IF (RK .EQ. 2) ALLUT = .TRUE.
C          IF (RK .EQ. 3) CX = .FALSE.
C          IF (RK .EQ. 4) SX = .FALSE.
C          IF (RK .EQ. 5) SPECIF = .FALSE.
C      ENDIF
C
C      IF (.NOT. (SAT .AND. CX .AND. SX)) ALSAT=.TRUE.
C
C 102 CALL FUNKY(KEYS(1,RK),RK)
C      IF (RK .EQ. 6) GO TO 200
C      GO TO 101
C
C 200 CONTINUE

```

```

REAL LOCLAT, LOCLNG, RADIUS
      LOCLAT - latitude of specified location
      LOCLNG - longitude of specified location
      RADIUS - maximum distance between specified location and
              file locations

COMMON /LOCN/ LOCLAT, LOCLNG, RADIUS

INTEGER IBASE(5), ILEV(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)

INTEGER SBUF(60), LBUF(90), MBUF(76), FBUF(50)

COMMON /BASE/ IBASE, ILEV, IMODE, ISTAT, LIST, ITDTE,
              ITTST, ITSAR, ITMCC, ITSPD, IDLUT, IDMCC,
              IDSAR, IDTST, SBUF, LBUF, MBUF, FBUF

DOUBLE PRECISION QTIME, SDT
      SDT - start date

      LODAT - ascii start date YYYYDD
      STDAT - low date in seconds from 1980
      ENDAT - high date in seconds from 1980

      OUTLU - output device, LU or filename

INTEGER RBUF(40), IPARM(5), PBUF(33)
INTEGER LU, RET
INTEGER SYS, ISYS, LOGLU

      RBUF - receiving buffer
      IPARM - return buffer for RMPAR call
      PBUF - return buffer from PARSE

REAL ONEDAY
INTEGER DATCH(3), LOOP, DUM(7), I1
INTEGER UNIT

LOGICAL LUT, MCC, SAR, TEST, ALL
      LUT - flag to indicate LUT location detail file selected
      MCC - flag to indicate MCC location detail file selected
      SAR - flag to indicate SARSTAT incident detail file selected
      TEST - flag to indicate FIELD test master file selected
      ALL - all files will be searched

INTEGER KEYS(8,8), PROM(8), RK
INTEGER FLG, AA, IFBRK

      FLG - flag to indicate if /E used
      AA - dummy parameter for IFBRK

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 IBUF(70),J,SEVENT,PASS(4),REGION,CASENO
INTEGER PLACE
REAL DIF

      ALLUT -flag to indicate if all locations wanted
      ALSAT -flag to indicate if all satellites wanted

      SAT -flag to indicate if specific satellites wanted
      SX -flag to indicate if all SARSAT satellites wanted
      CX -flag to indicate if all COSPAS satellites wanted
      SPESAT-array that contains the specific satellites desired

DATA KEYS // Specific SAT      A location  /
              , Cx              Sx          /
              , Satdas          END of select /
              , EXIT            /

DATA PROM //f1f2f3f4f5f6f7f8//
KEYS - array containing contents of soft keys
PROM - array of values to return for soft keys
RK - actual value returned

DATA IBUF1//TR,STPRIN, //
C*****

```

```

DATA ILEV /6HENTRY /
DATA LIST /040040B/
DATA ITDTE /6HDATE /
DATA ITTST /6HTEST /
DATA ITSAR /6HSARNDR/
DATA ITMCC /6HMCCREF/
DATA ITSPD /6HSATPAS/
DATA IDTST /6HFIELD /
DATA IDSAR /6HSARIF /
DATA IDMCC /6HMCCSELF/
DATA IDLUT /6HMLUTELF/
END

```

```

FTN4
$FILES(15,15)
PROGRAM LOCAT (3,99)
- SEF-40-00 V00 (840726.0907)
IMPLICIT NONE

```

SARSAT EVALUATION FACILITY

DATE	VERSION	DESCRIPTION	AUTHOR
83/10/05	00	-----	SUZANNE Y. SLINN

DESCRIPTION:

This program will retrieve data in a certain area during a specified time period. This program will compute distances between locations, and produce up to four output files depending on what the user specifies.

CALLING SEQUENCE:

RU,LOCAT

INPUTS:

OUTPUTS:

EXTERNAL REFERENCES:

Subroutines:

System:
RMPAR
PARSE
EXEC
IFBRK

User:

MOVBS - move a byte string
SESET - to query the operator for the date range, the device for the output file
FUNKY - to set soft keys (ZSCRN in ZSEFLB)
REILL - query operator for latitude and longitude
DOOPN - to open the database
DOCLS - to close the database

Functions:

GETCH - to get a character from a string
NUMBR - to convert ASCII string to integer
PTIME - to convert integer times to double precision seconds from 1980

DATA DECLARATIONS

```

COMMON /LOGG/ LODAT(3), MIDAT(3), STDAT, ENDAT, OUTLU(5), RANGE
INTEGER IOS
INTEGER LODAT, MIDAT, OUTLU, RANGE
DOUBLE PRECISION STDAT, ENDAT

```

FTM4

BLOCK DATA

 This is the block common for the program LOCAT

INTEGER LODAT, HIDAT, OUTLU, RANGE

LODAT - start date in ascii YYMMDD
 HIDAT - end date in ascii YYMMDD
 OUTLU - output device or file name
 RANGE - # of days to be processed

DOUBLE PRECISION STDAT, ENDAT

STDAT - start date in double precision seconds from 1980
 ENDAT - end date in double precision seconds from 1980

COMMON /LOGG/ LODAT(3), HIDAT(3), STDAT, ENDAT, OUTLU(5), RANGE

REAL LOCLAT, LOCLNG, RADIUS

LOCLAT - latitude of location specified
 LOCLNG - longitude of location specified
 RADIUS - maximum separation allowed for a match

COMMON /LOCN/ LOCLAT, LOCLNG, RADIUS

INTEGER IBASE(5), ILEV(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)

these are the parameters required by the IMAGE subroutines

IT--- is the ITEM parameter for each of the files accessed
 ID--- is the ID parameter for each of the files

INTEGER SBUF(68), LBUF(98), MBUF(76), FBUF(50)

SBUF - buffer for the SAR incident detail file
 LBUF - buffer for the LUT location detail file
 MBUF - buffer for the CMCC location detail file
 FBUF - buffer for the field test master file

COMMON /BASE/ IBASE, ILEV, IMODE, ISTAT, LIST, ITDTE,
 ITTST, ITSAR, ITMCC, ITSPD, IDLUT, IDMCC,
 IDSAR, IDTST, SBUF, LBUF, MBUF, FBUF

COMMON/SELECT/ALLUT, ALSAT, SAT, SX, CX, SPESAT, PRIM, SECOND, THIRD,
 FOURTH, PST, SST, TST, FST, UNIT, DIF, BENT, REALT, COS4, CBC12,
 CBC24, ALLFR, POSN, START, EXIT, TAPE, SPECIF, SEVENT, PASS,
 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

ALLUT - flag to indicate if all locations desired
 ALSAT - flag to indicate if all satellites desired
 SAT - flag to indicate that a specific satellite is desired
 SX - flag to indicate that all SRSAT satellites desired
 CX - flag to indicate that all COSPAS satellites desired
 SPESAT - 5 element array containing the names of the specific satellites desired
 PRIM - flag to indicate that the primary output was chosen
 SECOND - flag to indicate that the second output was chosen
 THIRD - flag to indicate that the third output was chosen
 FOURTH - flag to indicate that the fourth output was chosen
 PST - flag to indicate if the headers for the primary output have been displayed yet
 SST - flag to indicate if the headers for the second output have been displayed yet
 TST - flag to indicate if the headers for the third output have been displayed yet
 FST - flag to indicate if the headers for the fourth output have been displayed yet
 UNIT - variable containing the LU for the output
 DIF - variable containing the calculated difference between the calculated location and the true location
 BENT - flag to indicate that 406 'bent pipe' data desired
 REALT - flag to indicate that 406 2.4 Kb/s real time data desired
 COS4 - flag to indicate that 406 2.4 Kb/s COSPAS stored data desired
 CBC12 - flag to indicate that CBC121 data desired
 CBC24 - flag to indicate that CBC243 data desired
 ALLFR - flag to indicate that all the frequencies are desired
 EXIT - flag to indicate if the user wishes to abort the program
 TAPE - flag to indicate if output to go to magnetic tape
 SPECIF - flag to indicate if a specific SATPAS and EVENT desired
 SEVENT - event number of the specific SATPAS desired
 PASS - array containing the specific SATPAS desired

APPENDIX B

SOURCE CODE LISTINGS

THIRD OUTPUT
PRIMARY LOCATION

	DRIFT	CTAI	TCAT	MAJAX	MINAX	AMEAN	BIASI
1)	0.0000	20.0000	426.4746	84.862	36.784	0.0314	12500.
2)	0.0000	18.0000	548.4746	46.954	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	13.952	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	-0.2586	12000.
7)	0.0000	20.0000	318.4697	54.893	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	24.886	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	5.509	0.0142	12200.
15)	0.0000	3.0000	470.4746	17.860	6.816	-0.8824	9950.
16)	0.0188	6.0000	419.4746	56.680	15.028	0.0138	12600.
17)	0.0000	3.0000	409.4746	24.828	11.228	-3.0774	12800.
18)	0.0000	5.0000	423.4697	22.782	8.640	0.0240	10800.

FOURTH OUTPUT
PRIMARY LOCATION

	VARCTA	VARCTA	VARBTA	VARDT	CORCT	CORCB	CORCD	CORTB	CORTD	CORBD
1)	0.1533	4.3796	26.2943	0.0000	-0.6861	0.6179	0.0000	-0.7809	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	0.6053	-0.7221	-0.1229	-0.2186	-0.0996
4)	0.0391	0.3250	3.7128	0.0000	0.0966	-0.0699	0.0000	-0.9846	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	0.2105	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	0.6899	-0.6570	0.0000	-0.9982	0.0000	0.0000
8)	0.0566	0.3661	9.0059	0.8738	-0.0489	0.5309	-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	1.4395	7.5417	0.0000	-0.0754	0.1966	0.0000	-0.9573	0.0000	0.0000
12)	0.158	15.5563	81.4650	0.0000	-0.1952	0.8222	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	8.7565	19.8585	2.7615	0.0321	-0.5932	0.7662	-0.3588	0.0144	-0.8866
17)	0.0745	0.5669	7.2003	0.0000	0.1323	-0.4067	0.0000	-0.8597	0.0000	0.0000
18)	0.0649	0.6789	7.7781	0.0000	0.6928	-0.7947	0.0000	-0.9293	0.0000	0.0000

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, VJ - VICTORIA,
ENTER THEM ACCORDING TO THIS EXAMPLE: TR 1111

```

```

* Primary Output * Second Output * Third Output * Fourth Output
Type            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 10 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 transfer 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 (848726.0909)
IMPLICIT NONE

-----
SARSAT EVALUATION FACILITY
-----

DATE      VERSION  DESCRIPTION      AUTHOR
83/10/05   00      -----      SUZANNE Y. SLINN

DESCRIPTION:
  This subroutine determines which of the four output formats are
  desired.
CALLING SEQUENCE:
  CALLED FROM: LOCAT
  CALL OUTPT
-----

DATA DECLARATIONS
-----

COMMON /LOGG/ LODAT(3), HIDAT(3), STDAT, ENDAT, OUTLU(5), RANGE
INTEGER LODAT, HIDAT, OUTLU, RANGE
DOUBLE PRECISION STDAT, ENDAT

REAL LOCLAT, LOCLNG, RADIUS
  LOCLAT - latitude of specified location
  LOCLNG - longitude of specified location
  RADIUS - maximum distance between specified location and
          file locations

COMMON /LOCN/ LOCLAT, LOCLNG, RADIUS

INTEGER IBASE(5), ILEV(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)

INTEGER SBUF(68), LBUF(98), MBUF(76), FBUF(50)

COMMON /BASE/ IBASE, ILEV, IMODE, ISTAT, LIST, ITDTE,
- ITTST, ITSAR, ITMCC, ITSPD, IDLUT, IDMCC,
- IDSAR, IDTST, SBUF, LBUF, MBUF, FBUF

DOUBLE PRECISION QTIME, SDT
  SDT - start date

  LODAT - ascii start date YYYYMMDD
  STDAT - low date in seconds from 1980
  ENDAT - high date in seconds from 1980

  OUTLU - output device, LU or filename

INTEGER RBUF(40), IPARM(5), PBUF(33)
INTEGER LU, RET
INTEGER SYS, ISYS, LOGLU
  RBUF - receiving buffer
  IPARM - return buffer for RMPAR call
  PBUF - return buffer from PARSE

REAL ONEDAY
INTEGER DATCH(3), LOOP, DUM(7), I1
INTEGER UNIT

LOGICAL LUT, MCC, SAR, TEST, ALL
  LUT - flag to indicate LUT location detail file selected
  MCC - flag to indicate MCC location detail file selected
  SAR - flag to indicate SARSAT incident detail file selected
  TEST - flag to indicate FIELD test master file selected
  ALL - all files will be searched

```



```

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

```

```

FTN4
$FILES(15,15)
SUBROUTINE OUTP2(PLACE)
- SEF-40-00 V00 (840725.1426)
IMPLICIT NONE

```

```

-----
SARSAT EVALUATION FACILITY
-----
DATE      VERSION  DESCRIPTION      AUTHOR
83/10/05      00      -----      SUZANNE Y. SLINN
-----
DESCRIPTION:
  This subroutine opens the output files.
CALLING SEQUENCE:
  CALLED FROM: LOCAT
  CALL OUTPT
-----
DATA DECLARATIONS
-----

```

```

COMMON /LOGG/ LODAT(3), HIDAT(3), STDAT, ENDAT,OUTLU(5),RANGE
INTEGER PLACE
INTEGER LODAT, HIDAT, OUTLU, RANGE
DOUBLE PRECISION STDAT, ENDAT
REAL LOCLAT, LOCLNG, RADIUS
LOCLAT - latitude of specified location
LOCLNG - longitude of specified location
RADIUS - maximum distance between specified location and
         file locations
COMMON /LOCM/ LOCLAT, LOCLNG, RADIUS
INTEGER IBASE(5), ILEV(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
C      COMMON /BASE/ IBASE, ILEV, IMODE, ISTAT, LIST, ITDTE,
-      ITTST, ITSAR, ITMCC, ITSPD, IDLUT, IDMCC,
-      IDSAR, IDTST, SBUF, LBUF, MBUF, FBUF
C
C      DOUBLE PRECISION QTIME, SDT
C
C      SDT      - start date
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
C      INTEGER RBUF(40), IPARM(5), PBUF(33)
C      INTEGER LU, RET
C      INTEGER SYS, ISYS, LOGLU
C
C      RBUF     - receiving buffer
C      IPARM    - return buffer for RMPAR call
C      PBUF     - return buffer from PARSE
C
C      REAL ONEDAY
C      INTEGER DATCH(3), LOOP, DUM(7), I1
C      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 selected
C      ALL      - all files will be searched
C
C      INTEGER KEYS(8,8), PROM(8), RK
C      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
C      INTEGER CONTRY(3), IUSER, TYPE(4)
C      REAL ELTANG
C      LOGICAL ALLUT, ALSAT, SAT, SX, CX, PRIM, SECOND, THIRD, FOURTH
C      LOGICAL BENT, REALT, COS4, CBC12, CBC24, ALLFR, EXIT, TAPE, SPECIF
C      LOGICAL IMAGE
C      INTEGER SPESAT(5), I, IOS, PST, SST, TST, FST, UNIT, POSN, START
C      INTEGER SEVENT, PASS(4)
C      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 SRSAT satellites wanted
C      CX       -flag to indicate if all COSPAS satellites wanted
C      SPESAT   -array that contains the specific satellites desired
C
C      *****
C      EXECUTABLE CODE
C      *****
C
C      ISYS = LOGLU(SYS)
C
C      initialize variables
C
C      LU = 1
C      UNIT = 99
C
C      search files specified
C
C      100 CONTINUE
C      200 CONTINUE
C
C      OPEN(14, FILE='HEADR', IOSTAT=IOS, STATUS='OLD')
C      IF (IOS .NE. 0) WRITE(LU, 1000) IOS
C      IF (IOS .NE. 0) GO TO 999

```

```

C
open primary file if this output is desired
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
open second file if this output is desired
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
open third file if this output is desired
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
open fourth file if this output is desired
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
  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

```

```

FTM4
$FILES(15,15)
SUBROUTINE FREQ
- SEF-40-00 V00 (840726.0911)
IMPLICIT NONE

```

SARSAT EVALUATION FACILITY

DATE	VERSION	DESCRIPTION	AUTHOR
83/10/05	00	-----	SUZANNE Y. SLINN

DESCRIPTION:
This subroutine determines which of the frequencies the user desires to use.

CALLING SEQUENCE:

CALLED FROM: LOCAT
CALL FREQ

DATA DECLARATIONS

```

COMMON /LOGG/ LODAT(3), HIDAT(3), STDAT, ENDAT, OUTLU(5), RANGE
INTEGER LODAT, HIDAT, OUTLU, RANGE
DOUBLE PRECISION STDAT, ENDAT

```

```

REAL LOCLAT, LOCLNG, RADIUS

```

LOCLAT - latitude of specified location
LOCLNG - longitude of specified location
RADIUS - maximum distance between specified location and file locations

```

COMMON /LOCN/ LOCLAT, LOCLNG, RADIUS

```

```

INTEGER IBASE(5), ILEVEL(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)

```

```

INTEGER SBUF(60), LBUF(90), MBUF(76), FBUF(50)

```

```

COMMON /BASE/ IBASE, ILEVEL, IMODE, ISTAT, LIST, ITDTE,
ITTST, ITSAR, ITMCC, ITSPD, IDLUT, IDMCC,
IDSAR, IDTST, SBUF, LBUF, MBUF, FBUF

```

```

DOUBLE PRECISION QTIME, SDT

```

SDT - start date

LODAT - ascii start date YYMMDD
STDAT - low date in seconds from 1980
ENDAT - high date in seconds from 1980

OUTLU - output device, LU or filename

```

INTEGER RBUF(40), IPARM(5), PBUF(33)
INTEGER LU, RET
INTEGER SYS, ISYS, LOCLU

```

RBUF - receiving buffer
IPARM - return buffer for RMPAR call
PBUF - return buffer from PARSE

```

REAL ONEDAY
INTEGER DATCH(3), LOOP, DUM(7), I1
INTEGER UNIT

LOGICAL LUT, MCC, SAR, TEST, ALL

LUT - flag to indicate LUT location detail file selected
MCC - flag to indicate MCC location detail file selected
SAR - flag to indicate SARSTAT incident detail file selected
TEST - flag to indicate FIELD test master file selected
ALL - all files will be searched

INTEGER KEYS(8,6), PROM(8), RK
INTEGER FLG, AA, IFBRK

FLG - flag to indicate if /E used
AA - dummy parameter for IFBRK

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), I10S, PST, SST, TST, FST, UNIT, POSN, START
INTEGER SEVENT, PASS(4)
REAL DIF

ALLUT -flag to indicate if all locations wanted
ALSAT -flag to indicate if all satellites wanted
SAT -flag to indicate if specific satellites wanted
SX -flag to indicate if all SRSAT satellites wanted
CX -flag to indicate if all COSPAS satellites wanted
SPESAT -array that contains the specific satellites desired

DATA KEYS // 406 Bent Pipe 406 Real Time ',
- 406 Cospas CBC121 ',
- CBC243 END of select ',
- All frequencys EXIT '

DATA PROM // 'f1f2f3f4f5f6f7f8'//
KEYS - array containing contents of soft keys
PROM - array of values to return for soft keys
RK - actual value returned

*****
EXECUTABLE CODE
*****

ISYS = LOGLU(SYS)

initialize variables

ALLFR = .FALSE.
BENT = .FALSE.
REALT = .FALSE.
COS4 = .FALSE.
CBC12 = .FALSE.
CBC24 = .FALSE.
LU = 1
UNIT = 99

search files specified

100 CONTINUE
CALL FUNKY (KEYS, PROM)
WRITE (LU, 1003)
101 CALL REID(1, LU, RK, -2)
RK = RK - 2HFF
IF (RK .LT. 1 .OR. RK .GT. 8) GO TO 101
IF (RK .EQ. 8) THEN
EXIT = TRUE.
GO TO 999
ENDIF

set 8 in key selected and check for further selects

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) REAL= .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
C 102 CALL FUNKY(KEYS(1,RK),RK)
      IF (RK .EQ. 6) GO TO 200
      GO TO 101
C
C 200 CONTINUE
C
C      clear the '8' from the select array
C
C      DO 4 RK=1,8
          KEYS(1,RK)=2H
      CONTINUE
C
C 4
C
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      SANSAT EVALUATION FACILITY
C-----C
C      DATE      VERSION      DESCRIPTION      AUTHOR
C      83/10/05      00      -----      SUZANNE Y. SLINN
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      CALLING SEQUENCE:
C      CALLED FROM: LOCAT
C      CALL OUTDT
C-----C
C      DATA DECLARATIONS
C-----C
C      COMMON /LOGG/ LODAT(3), HIDAT(3), STDAT, ENDAT, OUTLU(5), RANGE
C      INTEGER LODAT, HIDAT, OUTLU, RANGE
C      DOUBLE PRECISION STDAT, ENDAT
C
C      REAL LOCLAT, LOCLNG, RADIUS, LUTLA, LUTLG, DSTMC
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-----C

```

```

COMMON /LOCN/ LOCLAT, LOCLNG, RADIUS
C
INTEGER IBASE(5), ILEV(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, ILEV, IMODE, ISTAT, LIST, ITDTE,
-           ITTST, ITSAR, ITMCC, ITSPD, IDLUT, IDMCC,
-           IDSAR, IDTST, SBUF, LBUF, MBUF, FBUF
C
DOUBLE PRECISION QTIME, SDT
C
SDT - start date
C
LODAT - ascii start date YYMMDD
SDAT - low date in seconds from 1980
ENDAT - high date in seconds from 1980
C
OUTLU - output device, LU or filename
C
INTEGER RBUF(40), IPARM(5), PBUF(33)
INTEGER LU, RET
INTEGER SYS, ISYS, LOGLU
C
RBUF - receiving buffer
IPARM - return buffer for RMPAR call
PBUF - return buffer from PARSE
C
REAL QMEDAY
INTEGER DATCH(3), LOOP, DUM(7), I1
INTEGER UNIT
C
LOGICAL LUT, MCC, SAR, TEST, ALL
C
LUT - flag to indicate LUT location detail file selected
MCC - flag to indicate MCC location detail file selected
SAR - flag to indicate SARSTAT incident detail file selected
TEST - flag to indicate FIELD test master file selected
ALL - all files will be searched
C
INTEGER KEYS(8,8), PROM(8), RK
INTEGER FLC, AA, IFBRK
C
FLC - flag to indicate if /E used
AA - dummy parameter for IFBRK
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
1
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
ALLUT -flag to indicate if all locations wanted
ALSAT -flag to indicate if all satellites wanted
C
SAT -flag to indicate if specific satellites wanted
SX -flag to indicate if all SAR5AT satellites wanted
CX -flag to indicate if all COSPAS satellites wanted
SPESAT -array that contains the specific satellites desired
C
DOUBLE PRECISION LATC, LATT, LONGT, LONGC, FA, DIFFR, DLAT, DLONG
REAL R, P
C
LATC -calculated latitude from LUT file
LATT -true latitude as input by the user
LONGT -true longitude as input by the user
LONGC -calculated longitude from the LUT file
DLAT -difference in latitude in km.
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 ELT1, ELT2, ELT3, ELT4, SPOT, ICONTR(10)
EQUIVALENCE(TRYA(1), MSG1(22))
EQUIVALENCE(TRYB(1), MSG1(27))
C
KEYS - array containing contents of soft keys
PROM - array of values to return for soft keys
RK - actual value returned
DATA ITEM/6HMCREF/

```

```

C *****
C EXECUTABLE CODE
C *****
C
C      SPOT = 0
C      CALL LGBUF(BGBUF,200)
C      ISYS = LOGLU(SYS)
C
C      initialize variables
C
C      read from the header file
C
C      REWIND (14)
C      READ(14,FMT=1022,IOSTAT=ISTAT)POS,FLAT,FLONG,FRADI
C      CPOS=POS
C      CRADI=FRADI
C      CLAT=FLAT
C      CLONG=FLONG
300 READ(14,FMT=1022,IOSTAT=ISTAT,END=121)POS,FLAT,FLONG,FRADI
C      IF (CPOS .EQ. POS) THEN
C          CPOS=POS
C          CRADI=FRADI
C          CLAT=FLAT
C          CLONG=FLONG
C          GO TO 300
C      ENDIF
C
C      IF (ISTAT .NE. 0) THEN
C          WRITE(1,120)ISTAT
120  FORMAT(' ERROR IN READING HEADER FILE, IOS IS ',I4)
C          GO TO 999
C      ENDIF
C      GO TO 122
121 POS=999
122 CONTINUE
C
C      PAGE=43
C      IF (CRADI .EQ. 0.0) THEN
C          LINE=7
C      ELSE
C          LINE=7
C      ENDIF
C      MLINE=1
C
C      IF (PRIM) THEN
C          P=3.141592654/180.
C          R=6378.
C          IF (PST .EQ. 1) GO TO 100
C          PST=1
C          WRITE(UNIT,1000)
C          WRITE(UNIT,1001)
C          WRITE(UNIT,1002)
C          WRITE(UNIT,1)
C          WRITE(UNIT,1023)CLAT,CLONG,CRADI
C          WRITE(UNIT,1)
100  CONTINUE
C          REWIND(10,IOSTAT=ISTAT,ERR=993)
C
C      initialize input buffer with blanks
C
C          I1=1
C          CALL FILBS(400,240,MSG1,I1)
C
C      read in record from primary file
C
101  CONTINUE
C      READ(10,FMT=1004,IOSTAT=ISTAT,ERR=993,END=103)(MSG1(I1),I1=1,50)
C
C      IF (ISTAT .NE. 0) THEN
C          WRITE(1,102)ISTAT
102  FORMAT(' READ ERROR, ISTAT IS ',I4)
C          GO TO 999
C      CONTINUE
C          WRITE(1,992)ISTAT
992  FORMAT(' ERROR IN READING FILE ',I4)
C      ENDIF
C      DECODE(9,1006,TRYA)LUTLA
C      DECODE(9,1006,TRYB)LUTLG
C
C      calculate delta lat, delta long
C
C      LATC=LUTLASP
C      LATI=CLATSP
C      LONGT=CLONGSP
C      LONGC=LUTLGSP
C
C      if a specific location was specified then
C      IF (CRADI .NE. 0.0) THEN
C          DIFFR=DSTNC(CLONG,CLAT,LUTLG,LUTLA)
C          BLAT=RS(LATC-LATI)
C          DLONG=RS(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
    RSPCE=PAGE-LINE
    DO 2 I1=1,RSPCE+8
        WRITE(UNIT,1)
        FORMAT(' ')
    2 CONTINUE
    WRITE(UNIT,1000)
    WRITE(UNIT,1001)
    WRITE(UNIT,1002)
    LINE=3
    NLINE=1
C
225 ENDIF
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
    RSPCE=PAGE-LINE
    DO 4 I1=1,RSPCE+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,1022)
    ELSE
        WRITE(UNIT,1009)
    ENDIF
104 CONTINUE
    REWIND(11)
    REWIND(14)
    READ(14,FMT=1022)POS,FLAT,FLONG,FRADI
    CPDS = POS
    READ(14,FMT=1022)POS,FLAT,FLONG,FRADI
C
C      initialize buffer with blanks
C
CALL FILRS(400,240,MSG1,11)
SPOT = SPOT + 1
C
C      read in record from second file
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
106      WRITE(1,106)ISTAT
      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)MLINE,(MSG1(I1),I1=1,52),
          1      CONTRY(1),CONTRY(2),IUSER,(TYPE(I1),I1=1,4)
        ELSE
          WRITE(UNIT,2021)MLINE,(MSG1(I1),I1=1,52),
          1      CONTRY(1),CONTRY(2),IUSER,ELTI2,ELTI3,ELTI4
        ENDIF
      ELSE
        WRITE(UNIT,1020)MLINE,(MSG1(I1),I1=1,52),ELTI1,ELTI2,ELTI3,
        1      ELTI4
      ENDIF
      LINE=LINE+1
      MLINE=MLINE+1
C
C      if end of page, skip to next page
      IF (LINE .GE. PAGE) THEN
        RSPGE=PAGE -LINE
        DO 5 I1=1,RSPGE+8
          WRITE(UNIT,I1)
        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
        MLINE=1
      ENDIF
C
C      SEE IF END OF QUERY
      IF (MLINE .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,I1)
        6      CONTINUE
        MLINE=1
        LINE=3
      ENDIF
C
C      if the third output is desired then output its headers
      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
C      initialize the buffer
      I1=1
      CALL FILDS(400,240,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
C 995 CONTINUE
        WRITE(1,110)ISTAT
C 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)
      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)
C 7 CONTINUE
          WRITE(UNIT,1011)
          WRITE(UNIT,1012)
          WRITE(UNIT,1013)
          LINE=3
          NLINE=1
C        ENDIF
        GO TO 109
      ENDIF
C 111 CONTINUE
      IF (FOURTH) THEN
        RSPGE=PAGE-LINE
        DO 8 I1=1,RSPGE+8
          WRITE(UNIT,1)
C 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)
C 113 CONTINUE
        REWIND(13)
C
C      initialize buffer with blanks
C
        I1=1
        CALL FILBS(400,240,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
C 997 CONTINUE
        WRITE(1,115)ISTAT
C 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)
      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)
C 9 CONTINUE
          WRITE(UNIT,1015)
          WRITE(UNIT,1016)
          WRITE(UNIT,1017)
          LINE=3
          NLINE=1
C        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
      NLINE=1
C   ENDIF
C   C
C   C***** OUTPUT IMAGE DATA *****
C   IF (IMAGE) THEN
C   C
C   C      initialize variables
C   C
      PST=0
      SST=0
      TST=0
      FST=0
C   C
C   C      read from the header file
C   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   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
      NLINE=1
C   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
C   C      initialize input buffer with blanks
C   C
      I1=1
      CALL FILBS(400,240,MSG1,I1)
C   C
C   C      read in record from primary file
C   C
701  CONTINUE
      READ(21,FMT=1004,IOSTAT=ISTAT,ERR=1993,END=703)
1    (MSG1(I1),I1=1,50)
C   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)LUTLC

```

```

C
C
C      compare to see if this is the correct record
      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
C      have the correct record
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
      I1=2
      CALL MOVBS(LBUF,9,6,MSG1,I1)
C
C      satpas & id
      I1=I1+1
      CALL MOVBS(LBUF,19,8,MSG1,I1)
C
C      Mccref
      CALL FORM(10,LBUF(4),MSG1,I1,6)
C
C      event
      CALL FORM(10,LBUF(3),MSG1,I1,6)
C
C      Messnt
      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
      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('          SECONDARY')
1002 FORMAT('    DATE SATPAS MCCREF EVENT MESSNT ELTLAT',
1          '    ELTLONG ELTLAT ELTLONG',
1          '    DIFF DLAT DLONG')
1004 FORMAT(50A2)
1005 FORMAT(' ***ERROR IN MCCREF, ISTAT IS ',I4)
      RETURN
      END

```



```

-----
SARSAT EVALUATION FACILITY
-----

DATE      VERSION      DESCRIPTION      AUTHOR
03/10/05      00      -----      SUZANNE Y. SLINN
-----

DESCRIPTION:
  This subroutine displays the first of the four output files on the
  desired logical unit.

CALLING SEQUENCE:
  CALLED FROM: ALOCS
  CALL PRIMR
-----

DATA DECLARATIONS
-----

COMMON /LOGG/ LODAT(3), HIDAT(3), STAT, ENDAT, OUTLU(5), RANGE
INTEGER LODAT, HIDAT, OUTLU, RANGE
DOUBLE PRECISION STAT, ENDAT

REAL LOCLAT, LOCLNG, RADIUS, LUTLA, LUTLG

LOCLAT - latitude of specified location
LOCLNG - longitude of specified location
RADIUS - maximum distance between specified location and
         file locations

COMMON /LOCN/ LOCLAT, LOCLNG, RADIUS

INTEGER IBASE(5), ILEV(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)

INTEGER SBUF(68), LBUF(98), MBUF(76), FBUF(50), BCBUF(500)

COMMON /BASE/ IBASE, ILEV, IMODE, ISTAT, LIST, ITDTE,
-             ITTST, ITSAR, ITMCC, ITSPD, IDLUT, IDMCC,
-             IDSAR, IDTST, SBUF, LBUF, MBUF, FBUF

DOUBLE PRECISION QTIME, SDT

SDT      - start date

LODAT    - ascii start date YYMMDD
STAT     - low date in seconds from 1980
ENDAT    - high date in seconds from 1980

OUTLU    - output device, LU or filename

INTEGER RBUF(40), IPARM(5), PBUF(33)
INTEGER LU, RET
INTEGER SYS, ISYS, LOGLU

RBUF     - receiving buffer
IPARM    - return buffer for RMPAR call
PBUF     - return buffer from PARSE

REAL OMEDAY
INTEGER BATCH(3), LOOP, DUM(7), II
INTEGER UNIT

LOGICAL LUT, MCC, SAR, TEST, ALL

LUT      - flag to indicate LUT location detail file selected
MCC      - flag to indicate MCC location detail file selected
SAR      - flag to indicate SARSTAT incident detail file selected
TEST     - flag to indicate FIELD test master file selected
ALL      - all files will be searched

```

```

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 DOCLS(IBASE,6HSARIF,1,ISTAT)
OPEN(30,FILE='SCRATCH',STATUS='OLD')
IF (SPECN EQ. 1HY) THEN
WRITE(30,1023)POSN,START,CONTRY(1)
ELSE
WRITE(30,1023)POSN,START,TYPE(1)
ENDIF
1023 FORMAT(15,1X,15,1X,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      Subroutine to break a single precision integer into its bit string.
C*****
C      SUBROUTINE BITS(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
S      DO 10 I = 1,15
J = 16 - I
IBIT(J) = 0
IF ((IB - 2**J) LT. 0) GO TO 10
IBIT(J) = 1
IB = IB - 2**J
10 CONTINUE
C      RETURN
END

```

```

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

```

```
DO 1 I=1,11  
IF (BEACON(I,1) .EQ. 2HCC) THEN  
TYPE(1) = 2MCG  
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. 2HR2) 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. 2HR5) 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. 2HDW) THEN  
TYPE(1) = 2HDN  
TYPE(2) = 2HDU  
TYPE(3) = 2HWI  
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
C determine if all COSPAS satellites are desired
C
C IF (CX) THEN
C   IF ((LBUF(10) .NE. 2HC1) .AND. (LBUF(10) .NE. 2HC2) .AND.
C     1 (LBUF(10) .NE. 2HC3) .AND. (LBUF(10) .NE. 2HC4)) GO TO 200
C   ENDIF
C
C determine if a specific satellite is desired
C
C IF (SAT) THEN
C   IF ((LBUF(10) .NE. SPESAT(1)) .AND. (LBUF(10) .NE. SPESAT(2))
C     1 .AND. (LBUF(10) .NE. SPESAT(3))
C     1 .AND. (LBUF(10) .NE. SPESAT(4))
C     1 .AND. (LBUF(10) .NE. SPESAT(5))) GO TO 200
C   ENDIF
C
C determine if a specific SATPAS is desired
C
C IF (SPECIF) THEN
C   IF ((LBUF(10) .NE. PASS(1)) .AND.
C     1 (LBUF(11) .NE. PASS(2)) .AND.
C     1 (LBUF(12) .NE. PASS(3)) .AND.
C     1 (LBUF(13) .NE. PASS(4))) GO TO 200
C   ENDIF
C
C determine which frequency of ELT is desired for output
C
C CHOSE=.FALSE.
C
C if not all the ELT frequencys are desired then
C
C IF (.NOT. (ALLFR)) THEN
C
C   if CBC121 and CBC243 are desired then
C
C   IF (CBC12 .AND. CBC24) THEN
C     IF ((LBUF(25) .NE. 8) .AND. (LBUF(25) .NE. 16)) GO TO 200
C     CHOSE = .TRUE.
C     GO TO 111
C   ENDIF
C
C   if CBC121 is desired then
C
C   IF (CBC12) THEN
C     IF (LBUF(25) .NE. 8) GO TO 200
C     CHOSE=.TRUE.
C   ENDIF
C
C   if CBC243 is desired then
C
C   IF (CBC24) THEN
C     IF (LBUF(25) .NE. 16) GO TO 200
C     CHOSE = .TRUE.
C   ENDIF
C
C   if 406 Kb/s real time is desired then
C
C   IF (REALT) THEN
C     IF (LBUF(25) .NE. 2) GO TO 200
C     CHOSE = .TRUE.
C   ENDIF
C
C   if 406 'bent pipe' data is desired then
C
C   IF (BENT) THEN
C     IF (LBUF(25) .NE. 1) GO TO 200
C     CHOSE=.TRUE.
C   ENDIF
C
C   if 406 2.4Kb/s COSPAS stored data is desired then
C
C   IF (COS4) THEN
C     IF (LBUF(25) .NE. 4) GO TO 200
C     CHOSE=.TRUE.
C   ENDIF
C
C   else an invalid option was chosen
C
C   IF (.NOT. (CHOSE)) THEN
C     564 WRITE(LU,564)
C     FORMAT(' AN ILLEGAL FREQUENCY CHOICE WAS MADE. DO AGAIN')
C     GO TO 999
C   ENDIF
C   ENDIF
C   111 CONTINUE

```

```

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

```

```

FTM4
$FILES(15,15)
PROGRAM ALOCS
C - SEF-48-88 V80 (848731.0886)
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, M, N
C
COMMON /LOCN/ LOCLAT, LOCLNG, RADIUS
C
INTEGER IBASE(5), ILEV(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, ILEV, IMODE, ISTAT, LIST, ITDTE,
ITTST, ITSAR, ITMCC, ITSPD, IDLUT, IDMCC,
IDSAR, IDTST, SBUF, LBUF, MBUF, FBUF
C
C
C
REAL ONEDAY, LUTLA, LUTLG, DIF, DSTMC
C
ONEDAY - counter of 24 hours
LUTLA - latitude on LUTELF record
LUTLG - longitude on LUTELF record
DIF - distance calculated
C
DOUBLE PRECISION QTIME, SDT
INTEGER LOOP, I, I1, DUM(7), DATE1(3)
C
LOOP - variable for do loop
I, I1 - counters
DUM - dummy buffer for call to QTIME
DATE1 - ascii date
C
INTEGER LU, FST, MSG1(40), UNIT
INTEGER*4 ALTSOL, INFO(7)
C
LU - terminal LU set to 1
FST - first time flag
MSG1 - output line buffer
UNIT - fortran LU set to 99
C
COMMON/SELECT/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), SPECOM, IBIT(16)
INTEGER REGION, CASENO, ICOUNR, IUSER, J, TYPE(4)
REAL ELTANG, PHI, AE, H, A, R1, CTA
INTEGER DEVENT, OLDSAT(4)
C
EQUIVALENCE (LUTLA, LBUF(20))
EQUIVALENCE (LUTLG, LBUF(30))
EQUIVALENCE (ALTSOL, LBUF(64))
EQUIVALENCE (CTA, LBUF(34))
C
C
C initialize variables
C
PHI = 3.14159265
AE = 6378.145
H = 1800.
C
OPEN(30, FILE='SCRATCH', STATUS='OLD')
READ (30, 1008) ALLUT, ALSAT, SAT, SX, CX, (SPESAT(I), I=1,5), PRIM,
1 SECOND, THIRD, FOURTH
1008 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      if end of page then skip to next page
C
C      IF (LINE .GE. PAGE) THEN
RSPGE=PAGE-LINE
DO 99 I=1,RSPGE+8
WRITE(UNIT,1)
99    CONTINUE
WRITE(UNIT,1015)
WRITE(UNIT,1016)
WRITE(UNIT,1017)
LINE=3
NLINE=1
C      ENDIF
ENDIF
GO TO 714
ENDIF
716 CONTINUE
C      IF (FOURTH) THEN
RSPGE=PAGE-LINE
DO 21 I=1,RSPGE+8
WRITE(UNIT,1)
21    CONTINUE
LINE=3
NLINE=1
C      ENDIF
ENDIF
1000 FORMAT('          PRIMARY DATA')
1001 FORMAT('          PRIMARY          SECONDARY')
1002 FORMAT('          DATE SATPAS MCCREF EVENT MESSMT ELTLAT',
1          '          ELTLONG DLAT ELTLAT ELTLONG',
1          '          DIFF DLAT DLONG')
1004 FORMAT(40A2,3(1X,F10.4))
1006 FORMAT(F10.3)
1007 FORMAT('          SECOND OUTPUT')
1008 FORMAT('          PRIMARY LOCATION')
1009 FORMAT('          CTA POINTS SDEV TREND QUAL PROB',
1          '          NMWLS TCA QTIME LOSTIN',
1          '          BIAS CORR SCORE')
1010 FORMAT(52A2,4(1X,I8))
1011 FORMAT('          THIRD OUTPUT')
1012 FORMAT('          PRIMARY LOCATION')
1013 FORMAT('          DRIFT CTAI TCAI MAJAX MINAX',
1          '          AWEAN BIASI')
1014 FORMAT(50A2)
1015 FORMAT('          FOURTH OUTPUT')
1016 FORMAT('          PRIMARY LOCATION')
1017 FORMAT('          VARCTA VARTCA VARBIA VARDI CORCT',
1          '          CORCB CORCD CORTB CORTD CORBD')
1018 FORMAT(50A2)
1019 FORMAT(1X,I3,'',3X,40A2,3(1X,F10.4))
1020 FORMAT(1X,I3,'',3X,52A2,4(1X,I8))
1021 FORMAT(1X,I3,'',3X,50A2)
1022 FORMAT(14,1X,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('          CTA POINTS SDEV TREND ELTLANG PROB',
1          '          NMWLS TCA QTIME LOSTIN',
1          '          BIAS ELT ID')
999 CONTINUE
RETURN
END

```

```

C
C      if the third output is desired then output its headers
C      IF (THIRD) THEN
C          IF (TST .EQ. 1) GO TO 708
C          TST=1
C          WRITE(UNIT,1011)
C          WRITE(UNIT,1012)
C          WRITE(UNIT,1013)
708      CONTINUE
C          REWIND(23)
C
C          initialize the buffer
C          I1=1
C          CALL FILBS(400,240,MSG1,I1)
C
C          read in record from third file
C      709      CONTINUE
C          read a record from the third output file
C          READ(23,1014,IOSTAT=ISTAT,ERR=1995,END=1111)(MSG1(I1),I1=1,50)
C          IF (ISTAT .NE. 0) THEN
1995      CONTINUE
C          WRITE(1,710)ISTAT
710      FORMAT(' READ ERROR ON THIRD FILE, ISTAT IS ',I4)
C          GO TO 999
C      ENDIF
C          write a record onto the output logical unit
C          WRITE(UNIT,1021)NLINE,(MSG1(I1),I1=1,50)
C          NLINE=NLINE+1
C          NLINE=NLINE+1
C
C          if end of page then skip to the next page
C      IF (LINE .GE. PAGE) THEN
C          RSPGE=PAGE-LINE
C          DO 777 I1=1,RSPGE+8
C              WRITE(UNIT,1)
777      CONTINUE
C          WRITE(UNIT,1011)
C          WRITE(UNIT,1012)
C          WRITE(UNIT,1013)
C          LINE=3
C          NLINE=1
C      ENDIF
C          GO TO 709
C      ENDIF
1111 CONTINUE
C      IF (FOURTH) THEN
C          RSPGE=PAGE-LINE
C          DO 888 I1=1,RSPGE+8
C              WRITE(UNIT,1)
888      CONTINUE
C          LINE=3
C          NLINE=1
C      ENDIF
C
C      if the fourth output is desired then display its headers
C      IF (FOURTH) THEN
C          IF (FST .EQ. 1) GO TO 713
C          FST=1
C          WRITE(UNIT,1015)
C          WRITE(UNIT,1016)
C          WRITE(UNIT,1017)
713      CONTINUE
C          REWIND(24)
C
C          initialize buffer with blanks
C          I1=1
C          CALL FILBS(400,240,MSG1,I1)
C
C          read in record from fourth file
C      714      CONTINUE
C          read a record from the fourth output file
C          READ(24,1018,IOSTAT=ISTAT,ERR=1997,END=716)(MSG1(I1),I1=1,50)
C          IF (ISTAT .NE. 0) THEN
1997      CONTINUE
C          WRITE(1,715)ISTAT
715      FORMAT(' READ ERROR ON FOURTH FILE, ISTAT IS ',I4)
C          GO TO 999
C      ENDIF
C          write a record from the output file
C          WRITE(UNIT,1021)NLINE,(MSG1(I1),I1=1,50)
C          NLINE=NLINE+1
C          NLINE=NLINE+1

```

```

C
C
C      initialize buffer with blanks
      I1=1
      CALL FILBS(40B,240,MSG1,I1)
C
C      read in record from second file
      SPOT = SPOT + 1
705  CONTINUE
      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)MLINE,(MSG1(I1),I1=1,52),CONTRY(1),
      1  CONTRY(2),IUSER,(TYPE(I1),I1=1,4)
      ELSE
      WRITE(UNIT,2021)MLINE,(MSG1(I1),I1=1,52),CONTRY(1),
      1  CONTRY(2),IUSER,ELTI2,ELTI3,ELTI4
      ENDIF
      ELSE
      WRITE(UNIT,1020)MLINE,(MSG1(I1),I1=1,52),ELTI1,ELTI2,ELTI3,
      1  ELTI4
      ENDIF
      LINE=LINE+1
      MLINE=MLINE+1
C
C
C      if end of page, ship to next page
      IF (LINE GE. PAGE) THEN
      RSPGE=PAGE-LINE
      DO 2555 I1=1,RSPGE+8
      WRITE(UNIT,I1)
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
      MLINE=1
C      ENDIF
C
C      SEE IF END OF QUERY
C
C      IF (MLINE 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
      RSPGE=PAGE-LINE
      DO 666 I1=1,RSPGE+8
      WRITE(UNIT,I1)
666  CONTINUE
      MLINE=1
      LINE=3
      ENDIF

```

```

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

```

```

FTN4
$FILES(1,1)
SUBROUTINE SECXY(OTHDAT,SPECN)
- SEF-40-00 V00 (840727.1241)
IMPLICIT NONE

C-----
C
C SARGAT EVALUATION FACILITY
C-----
C
C DATE      VERSION  DESCRIPTION      AUTHOR
C 83/10/85   00      -----      SUZANNE Y. SLINN
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 SECXY
C-----
C
C DATA DECLARATIONS
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 /LOCH/ LOCLAT, LOCLNG, RADIUS
C
INTEGER IBASE(5), ILEV(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(60), LBUF(90), MBUF(76), FBUF(50)
C
COMMON /BASE/ IBASE, ILEV, IMODE, ISTAT, LIST, ITDTE,
- ITTST, ITSAR, ITMCC, ITSPD, IDLUT, IDMCC,
- IDSAR, IDTST, SBUF, LBUF, MBUF, FBUF
C
DOUBLE PRECISION QTIME, SDT
C
SDT - start date
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, LOCLU
C
RBUF - receiving buffer
IPARM - return buffer for RMPAR call
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
LUT - flag to indicate LUT location detail file selected
MCC - flag to indicate MCC location detail file selected
SAR - flag to indicate SARSTAT incident detail file selected
TEST - flag to indicate FIELD test master file selected
ALL - all files will be searched

```



```

      number of WLS iterations (NMWLS)
      I1=I1+1
      CALL FORM(10,LBUF(43),MSG1,I1,6)
      time of closest approach in hours from the date of AOS (TCA)
      I1=I1+1
      CALL FPNBS(LBUF(36),177744B,MSG1,I1)
      time of AOS (QTIME)
      I1=I1+4
      CALL FPNBS(LBUF(8),177744B,MSG1,I1)
      date of LOS, in hours from date of AOS (LOSTIM)
      I1=I1+4
      CALL FPNBS(LBUF(19),177744B,MSG1,I1)
      ELT frequency bias, expressed Hz (BIAS)
      ENCODE(20,123,TRY)BIAS
123  FORMAT(F7.0)
      Elt id is written to the file from LBUF(21),LBUF(22),LBUF(23),LBUF(24)

      IF 406 DATA, CHANGE THE ELTID CODE OUTPUT
      IF ((BENT) .OR. (REALT) .OR. (COS4)) THEN
        IF COUNTRY IS CANADA, OUTPUT BEACON NAME
        IF ((CONTRY(1) .EQ. 2NCA) .AND. (SPECOM .EQ. 1HY)) THEN
          IF (QTHDAT) 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 (QTHDAT) 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 (QTHDAT) 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,13,1X,4A2)
      1005 FORMAT(52A2,2A2,1X,13,3(1X,I8))
      1006 FORMAT(52A2,4(1X,I8))
      RETURN
      END

```

```

FTN4
@FILES(1,1)
SUBROUTINE TERTI(OTMIDAT)
- SEF-40-88 V88 (840726.0984)
IMPLICIT NONE

```

SARSAT EVALUATION FACILITY

DATE	VERSION	DESCRIPTION	AUTHOR
83/10/05	00	---	SUZANNE Y. SLINN

DESCRIPTION:

This subroutine displays the third of the four output files.

CALLING SEQUENCE:

CALLED FROM: ALOCS

CALL TERTI

DATA DECLARATIONS

```

COMMON /LOGG/ LODAT(3), HIDAT(3), STDAT, ENDAT, OUTLU(5), RANGE
INTEGER LODAT, HIDAT, OUTLU, RANGE, TRY(10)
DOUBLE PRECISION STDAT, ENDAT

```

```

REAL LOCLAT, LOCLNG, RADIUS, LUTLA, LUTLG, BIAS

```

```

LOCLAT - latitude of specified location
LOCLNG - longitude of specified location
RADIUS - maximum distance between specified location and
file locations

```

```

COMMON /LOCN/ LOCLAT, LOCLNG, RADIUS

```

```

INTEGER IBASE(5), ILEV(3), INODE, ISTAT(10), LIST
INTEGER ITDTE(3), ITTST(3), ITSAR(3), ITMCC(3), ITSPD(3)
INTEGER IDLUT(3), IDTST(3), IDSAR(3), IDMCC(3)

```

```

INTEGER SBUF(60), LBUF(90), MBUF(70), FBUF(50)

```

```

COMMON /BASE/ IBASE, ILEV, INODE, ISTAT, LIST, ITDTE,
ITTST, ITSAR, ITMCC, ITSPD, IDLUT, IDMCC,
IDSAR, IDTST, SBUF, LBUF, MBUF, FBUF

```

```

DOUBLE PRECISION QTIME, SDT

```

SDT - start date

```

LODAT - ascii start date YYMMDD
STDAT - low date in seconds from 1980
ENDAT - high date in seconds from 1980

```

OUTLU - output device, LU or filename

```

INTEGER RBUF(40), IPARM(5), PBUF(33)
INTEGER LU, RET
INTEGER SYS, ISYS, LOGLU

```

```

RBUF - receiving buffer
IPARM - return buffer for RMPAR call
PBUF - return buffer from PARSE

```

```

REAL ONEDAY
INTEGER BATCH(3), LOOP, DUM(7), I1
INTEGER UNIT

```

```

LOGICAL LUT, MCC, SAR, TEST, ALL

```

```

LUT - flag to indicate LUT location detail file selected
MCC - flag to indicate MCC location detail file selected
SAR - flag to indicate SARSAT incident detail file selected
TEST - flag to indicate FIF:0 test master file selected
ALL - all files will be searched

```

```

INTEGER KEYS(8,8), PROM(8), RK
INTEGER FLC, AA, IFBRK

```

```

      FLC - flag to indicate if /E used
      AA - dummy parameter for IFBRK

```

```

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), PST, SST, TST, FST, MSG1(80), POSN, START
INTEGER SEVENT, PASS(4)
LOGICAL IMAGE, OTHDAT
REAL DIF

```

```

      ALLUT -flag to indicate if all locations wanted
      ALSAT -flag to indicate if all satellites wanted

```

```

      SAT -flag to indicate if specific satellites wanted
      SX -flag to indicate if all SARSAT satellites wanted
      CX -flag to indicate if all COSPAS satellites wanted
      SPESAT-array that contains the specific satellites desired

```

```

EQUIVALENCE(LUTLA, LBUF(28))
EQUIVALENCE(LUTLG, LBUF(30))
EQUIVALENCE(BIAS, LBUF(78))
EQUIVALENCE(TRY, MSG1(35))

```

```

      KEYS - array containing contents of soft keys
      PROM - array of values to return for soft keys
      RK - actual value returned

```

``` ***** EXECUTABLE CODE ***** ```

```

      ISYS = LOGLU(SYS)

```

```

      initialize variables

```

```

100 CONTINUE

```

```

      Ii=1
      CALL FILBS(40B, 160, MSG1, Ii)

```

```

      drift

```

```

      Ii=2
      CALL FPNBS(LBUF(40), 177744B, MSG1, Ii)

```

```

      initial estimate of CTA in degrees (CTAI)

```

```

      Ii=Ii+1
      CALL FPNBS(LBUF(66), 177744B, MSG1, Ii)

```

```

      initial estimate of TCA in seconds (TCAI)

```

```

      Ii=Ii+1
      CALL FPNBS(LBUF(68), 177744B, MSG1, Ii)

```

```

      initial estimate of BIAS in Hz (BIASI)

```

```

123 ENCODE(28, 123, TRY)BIAS
      FORMAT(F7.0)

```

```

      major axis of error ellipse in Km. (MAJAX)

```

```

      Ii=Ii+1
      CALL FPNBS(LBUF(55), 177743B, MSG1, Ii)

```

```

      minor axis of error ellipse in Km. (MINAX)

```

```

      Ii=Ii+1
      CALL FPNBS(LBUF(57), 177743B, MSG1, Ii)

```

```

      average of data residuals in Hz (AMEAN)

```

```

      Ii=Ii+5
      CALL FPNBS(LBUF(72), 177744B, MSG1, Ii)

```

```

      IF (OTHDAT) THEN
        WRITE(23, 1004)(MSG1(Ii), Ii=1, 50)

```

```

      ELSE
        WRITE(12, 1004)(MSG1(Ii), Ii=1, 50)

```

```

      ENDIF

```

```

1004 FORMAT(50A2)
      RETURN
      END

```

```

FTN4
$FILES(2,2)
SUBROUTINE QUADY(OTHDAT)
- SEF-40-00 V00 (840726.0905)
IMPLICIT NONE

```

SARSAT EVALUATION FACILITY

DATE	VERSION	DESCRIPTION	AUTHOR
03/10/05	00	-----	SUZANNE Y. SLINN

DESCRIPTION:
This subroutine displays the fourth of the four output files.

CALLING SEQUENCE:

CALLED FROM: ALOCS
CALL QUADY

DATA DECLARATIONS

```

COMMON /LOGG/ LODAT(3), HIDAT(3), STDAT, ENDAT, OUTLU(5), RANGE
INTEGER LODAT, HIDAT, OUTLU, RANGE, TRY(10)
DOUBLE PRECISION STDAT, ENDAT

```

```

REAL LOCLAT, LOCLNG, RADIUS, LUTLA, LUTLG

```

LOCLAT - latitude of specified location
LOCLNG - longitude of specified location
RADIUS - maximum distance between specified location and file locations

```

COMMON /LOCN/ LOCLAT, LOCLNG, RADIUS

```

```

INTEGER IBASE(5), ILEV(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)

```

```

INTEGER SBUF(60), LBUF(90), MBUF(76), FBUF(50)

```

```

COMMON /BASE/ IBASE, ILEV, IMODE, ISTAT, LIST, ITDTE,
- ITTST, ITSAR, ITMCC, ITSPD, IDLUT, IDMCC,
- IDSAR, IDTST, SBUF, LBUF, MBUF, FBUF

```

```

DOUBLE PRECISION QTIME, SDT

```

SDT - start date

LODAT - ascii start date YYYYDD
STDAT - low date in seconds from 1980
ENDAT - high date in seconds from 1980

OUTLU - output device, LU or filename

```

INTEGER RBUF(40), IPARM(5), PBUF(33)
INTEGER LU, RET
INTEGER SYS, ISYS, LOGLU

```

RBUF - receiving buffer
IPARM - return buffer for RHPAR call
PBUF - return buffer from PARSE

```

REAL ONEDAY
INTEGER DATCH(3), LOOP, DUM(7), I1
INTEGER UNIT

```

```

LOGICAL LUT, MCC, SAR, TEST, ALL

```

LUT - flag to indicate LUT location detail file selected
MCC - flag to indicate MCC location detail file selected
SAR - flag to indicate SARSTAT incident detail file selected
TEST - flag to indicate FIELD test master file selected
ALL - all files will be searched

INTEGER KEYS(8), PROM(8), RK
 INTEGER FLC, AA, IFBRK

FLC - flag to indicate if /E used
 AA - dummy parameter for IFBRK

COMMON/SELECT/ALLUT, ALSAT, SAT, SX, CX, SPESAT, PRIM, SECOND, THIRD,
 FOURTH, PST, SST, TST, FST, UNIT, DIF, BENT, REALT, COS4, CBC12,
 CBC24, ALLFR, POSN, START, EXIT, TAPE, SPECIF, SEVENT, PASS,
 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

ALLUT -flag to indicate if all locations wanted
 ALSAT -flag to indicate if all satellites wanted

SAT -flag to indicate if specific satellites wanted
 SX -flag to indicate if all SIRSAT satellites wanted
 CX -flag to indicate if all COSPAS satellites wanted
 SPESAT-array that contains the specific satellites desired

EQUIVALENCE(LUTLA, LBUF(28))
 EQUIVALENCE(LUTLG, LBUF(30))

KEYS - array containing contents of soft keys
 PROM - array of values to return for soft keys
 RK - actual value returned

EXECUTABLE CODE

ISYS = LOGLU(SYS)

initialize variables

100 CONTINUE

I1=1

CALL FILBS(400, 160, MSG1, I1)

standard deviation of CTA in degrees (VARCTA)

I1=2

CALL FPNBS(LBUF(78), 1777440, MSG1, I1)

standard deviation of TCA in seconds (VARTCA)

I1=I1+1

CALL FPNBS(LBUF(80), 1777440, MSG1, I1)

standard deviation of BIAS in Hz (VARBIA)

I1=I1+1

CALL FPNBS(LBUF(82), 1777440, MSG1, I1)

standard deviation of DRIFT in Hz/min (VARDR1)

I1=I1+1

CALL FPNBS(LBUF(84), 1777440, MSG1, I1)

correlation of CTA with TCA (CORCT)

I1=I1+1

CALL FPNBS(LBUF(86), 1777440, MSG1, I1)

correlation of CTA with BIAS (CORCB)

I1=I1+1

CALL FPNBS(LBUF(88), 1777440, MSG1, I1)

correlation of CTA with DRIFT (CORCD)

I1=I1+1

CALL FPNBS(LBUF(90), 1777440, MSG1, I1)

correlation of TCA with BIAS (CORTB)

I1=I1+1

CALL FPNBS(LBUF(92), 1777440, MSG1, I1)

correlation of TCA with DRIFT (CORTD)

I1=I1+1

CALL FPNBS(LBUF(94), 1777440, MSG1, I1)

```
C      correlation of BIAS with DRIFT (CORBD)
C
      I1=I1+1
      CALL FPNBS(LBUF(96),1777448,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
```



```

      INTEGER DEV, PTRLU, IPARM(5), IBUF(30)
      INTEGER SYS, ISYS, LOGLU

C
C  DEV      - Terminal LU
C  PTRLU    - Printer LU Set to 6
C  IPARM    - Buffer for RPAR
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
      DATA NO, YE, SLSHE, END / 2HNO, 2HYE, 275058, 2HEN /
      DATA DAYS / 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 /

C
C  =====
C  INITIALIZATION!!
C  =====
C
      ISYS = LOGLU(SYS)
      DEV=1
      RET=0
      PTRLU=6

C
C  =====
C
C  Get current date and time from the system
C  (numeric)
C
      CALL EXEC(I1, RBUF, DBUF(1))
      DBUF(2)=0
      CALL DATE(DBUF(1), DBUF(2), RBUF(5))
      DBUF(1)=DBUF(1)-1900
      DBUF(3)=RBUF(5)

```



```

C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  EXIT!!
C
  900 CONTINUE
    RETURN
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  FORMATS:
C
  9000 FORMAT('Enter date for start and end of search: ',
    &         'to TYNMDD TYNMDD')
  9001 FORMAT('Specify retrieval output device LU: _')
  9002 FORMAT(39A2)
  9003 FORMAT('  ERROR 8 File ",J2," ",06," ",06," access error "
    &         ",16," ')
  9004 FORMAT('  ERROR 8 Answer YES or NO ')
  9005 FORMAT('  ERROR 8 Disaster called incorrectly!')
  9007 FORMAT('Enter date for stop of search ')
  9008 FORMAT('  ERROR 8 Year incorrect ')
  9009 FORMAT('  ERROR 8 Month incorrect ')
  9010 FORMAT('  ERROR 8 Day incorrect ')
  9011 FORMAT('  ERROR 8 High date precedes low date')
  9012 FORMAT(3I2,1X,3I2)
    END

```

##This file writes the output files onto the magnetic tape

##

SV.1.,IM

ST,10.B

IF,20.EQ./E,5

ST,20.B

IF,30.EQ./E,3

ST,30.B

IF,40.EQ./E,1

ST,40.B

IF,50.EQ./E,1

ST,50.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.LR
RE.ZLOCAT
RE.ZLOCBL
**RE.ZALDCS
RE.ZOUTPUT
RE.ZOUTP2
**RE.ZPRIMARY
**RE.ZSECOND
**RE.ZTHIRD
**RE.ZFOURTH
RE.ZFREQUENCY
RE.ZOUTDAT
RE.ZSEHCC
RE.ZSESAR
RE.ZSETST
RE.ZRETLI
RE.ZSESE1
RE.ZDSTNC
RE.ZCLONS
SE
SE.ZDOLIB
SE.ZDRASE
SE.ZLTSB3
SEA.ZSEFLB
SEA.ZRULIB
/E

```

```

*****
**
**      PROCEDURE :ALDCS
**
**      14 MAY, 1984
**
**      PROGRAMMER SUZANNE Y. SLINN
**
**      VERSION      1.0      (840515.1348)
**
*****
**
**      DESCRIPTION
**
**      Load sequence for ALDCS
**
*****

```

```

EC
SZ.29
OP.LR
RE.ZLOCBL
RE.ZALDCS
RE.ZOUTP2
RE.ZPRIMARY
RE.ZSECOND
RE.ZTHIRD
RE.ZFOURTH
**RE.ZFREQUENCY
**RE.ZOUTDAT
RE.ZSEHCC
RE.ZSESAR
RE.ZSETST
RE.ZRETLI
RE.ZSESE1
RE.ZDSTNC
RE.ZCLONS
SE
SE.ZDOLIB
SE.ZDRASE
SE.ZLTSB3
SEA.ZSEFLB
SEA.ZRULIB
/E

```

```

LOADR, LOCAT
SZ, 29
OP, LB
RE, ZLOCAT
LOCAT 10042 21126 SEF-40-00 V00 (840727.1242)
RE, ZLOCBL
SELECT 21127 21206
BASE 21207 21731
LOCN 21732 21737
LOGG 21740 21761
**RE, ZALOGS
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, ZSEMC
SEMC 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, ZRETL
RETL 41630 43142 SEF-40-00 V00 (830526.1756)
RE, ZSESE1
SESE1 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 750781
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 800719
MESSS 47662 50215 92067-16261 REV 1903 790420
PARSE 50216 50235 92067-1X281 REV 2013 770714
CNUMD 50236 50255 92067-1X284 REV 2013 770621
ERR0 50256 50263 24998-1X250 REV 2140 810506
JID 50264 50437 24998-1X343 REV 2140 810422
IFTTY 50440 50525 92067-1X295 REV 2013 790118
CAPCK 50526 51052 92067-1X310 REV 2013 790801
QSHVE 51053 51145 92067-1X403 REV 2013 800129
SESTB 51146 51162 92067-1X454 REV 2013 790202
SESSM 51163 51200 92067-16125 REV 1903 780413
REID 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
EID 53501 53553 24998-1X329 REV 2140 810422
FMEN 53554 53635 24998-1X345 REV 2140 810415
IDER 53636 54003 24998-1X321 REV 2140 810506
FMFP 54004 55247 24998-1X346 REV 2140 810519
FMOT 55248 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
FMID 57748 60171 24998-1X348 REV 2140 810406
IOOP 60172 60200 24998-1X300 REV 2101 800805
IOCN 60201 60244 24998-1X327 REV 2101 801007
FFOP 60245 61366 24998-1X301 REV 2140 810724 N
CREAT 61367 61751 92067-16125 REV 2140 810629
FOP1 61752 62027 24998-1X326 REV 2101 800729
FFCL 62028 62361 24998-1X306 REV 2140 810720 N
LOCF 62362 62662 92067-16125 REV 1903 781110
RWDF 62663 62747 92067-16125 REV 1903 780724
POST 62748 62776 92067-16125 REV 1903 740801

```

DCO	62777	63047	24998-1X037	REV. 2001	790417
DDI	63050	63350	24998-1X040	REV. 2001	781021
LOCN	63351	63357	24998-1X315	REV. 2101	800731
FFCN	63360	63610	24998-1X316	REV. 2140	800813 M
POSMT	63611	64151	92067-16125	REV. 1903	790316
FTOI	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
DIM	64531	64536	24998-1X042	REV. 2001	780818
FCM	64537	64553	24998-1X102	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	24998-1X168	REV. 2001	750801
ASIN	65263	65366	24998-1X373	REV. 2101	800222
DNG	65367	65376	24998-1X046	REV. 2001	780818
DADS	65377	65506	24998-1X036	REV. 2001	780818
DIS	65507	65536	24998-1X044	REV. 2001	780818
QVRD	65537	65537	92067-16125	REV. 1903	780526
IDRP	65540	67143	92067-16185	REV. 2026	800305
R/WB	67144	67302	92067-16125	REV. 2101	801013
\$OPEN	67303	67457	92067-16125	REV. 1903	790103
RWUR	67460	70031	92067-16125	REV. 2101	800303
RWWD*	70032	70161	92067-16125	REV. 2101	810617
VSCBA	70162	70227	92067-16261	REV. 1903	790202
\$PARS	70230	70450	92067-1X282	REV. 2013	780811
\$CUT3	70451	70536	92067-1X287	REV. 2013	770621
LURQ	70537	71151	92067-1X270	REV. 2013	791024
NAR	71152	71246	92067-16125	REV. 1903	740801
P. PAS	71247	71275	92067-16125	REV. 1903	740801
LUTRU	71276	71404	92067-1X308	REV. 2013	790223
\$ALRN	71405	71522	92067-1X271	REV. 2013	770715
QMMR	71523	71570	92067-1X324	REV. 2013	780919
LINEH	71571	71611	92067-1X477	REV. 2013	790126
PNAME	71612	71662	92068-1X035	REV. 2101	800919
OPN?	71663	71706	24998-1X325	REV. 2101	800803
ERO. 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. XDOLIB					
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. XSEFLB					
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)
FURN	74746	75265	Format interder numbers	V01	(840404.0945)
NUMBR	75266	75440	Decode interder 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
AINI	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: RP PROGRAM: LB LOAD: TE COMMON: NC
 /LOADR: LOCAT READY AT 10:31 AM MON., 17 SEPT, 1984

/LOADR: SEND

LOADR:ALOCS

SZ,29
 OP,LB
 RE,ZLOCR
 SELECT 10042 10121
 RASE 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
 SEC0Y 26555 30332 SEF-40-00 V00 (840727.1241)
 RE,ZTMR0
 TERT1 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,ZSETSI
 SETSI 33770 34505 SEF-40-00 V00 (830908.1401)
 RE,ZRETL
 RETLL 34506 36020 SEF-40-00 V00 (830526.1756)
 RE,ZSESEI
 SESEI 36021 37577 SEF -040-00 V00 (840514.0857)
 RE,ZDSTMC
 DSTMC 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
 IDRP1 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 92067-1X201 REV.2013 770714
 CMURD 43114 43133 92067-1X204 REV.2013 770621
 ERRO 43134 43141 24998-1X250 REV.2140 810506
 JTO 43142 43315 24998-1X343 REV.2140 810422
 JTTY 43316 43403 92067-1X295 REV.2013 790110
 CAPCX 43404 43730 92067-1X310 REV.2013 790801
 %SMVE 43731 44023 92067-1X403 REV.2013 800129
 %ESTB 44024 44040 92067-1X454 REV.2013 790202
 %SSN 44041 44056 92067-16125 REV.1903 780413
 LGBUF 44057 44075 24998-1X362 REV.2140 810505
 FMGB 44076 44351 24998-1X353 REV.2140 810420
 IOCL 44352 44453 24998-1X305 REV.2101 800731
 EXIT 44454 44527 24998-1X320 REV.2101 800731
 FFRW 44530 44626 24998-1X297 REV.2140 810422 M
 READF 44627 46136 92067-16125 REV.2101 810616
 FMR 46137 46203 24998-1X352 REV.2140 810515
 EIO 46204 46256 24998-1X329 REV.2140 810422
 FMCH 46257 46340 24998-1X345 REV.2140 810415
 IOER 46341 46506 24998-1X321 REV.2140 810506
 FMFP 46507 47752 24998-1X346 REV.2140 810519
 FMOT 47753 50034 24998-1X351 REV.2140 810415
 UFMF 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 24998-1X300 REV.2101 800805
 IOCH 53011 53054 24998-1X327 REV.2101 801007
 FFOP 53055 54176 24998-1X301 REV.2140 810724 M
 CREAT 54177 54501 92067-16125 REV.2140 810629
 FOP? 54502 54637 24998-1X326 REV.2101 800729
 FFCL 54640 55171 24998-1X306 REV.2140 810728 M
 LOCF 55172 55472 92067-16125 REV.1903 781110
 RWMD 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
 RTOI 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 24998-1X182 REV.2001 750701

.ASIN 56782 57885 24998-1X373 REV. 2101 800222
 .DNC 57886 57815 24998-1X046 REV. 2001 788818
 .DADS 57816 57125 24998-1X036 REV. 2001 788818
 .OVRD. 57126 57126 92067-16125 REV. 1903 780526
 .IDRP 57127 60532 92067-16185 REV. 2026 800305
 .R/US 60533 60671 92067-16125 REV. 2101 801013
 .OPEN 60672 61046 92067-16125 REV. 1903 790103
 .RWSUB 61047 61420 92067-16125 REV. 2101 800303
 .RWND9 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 92067-1X278 REV. 2013 791024
 .NAM. 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
 .LIMEH 63160 63280 92067-1X477 REV. 2013 790126
 .REIO 63281 63325 92067-1X275 REV. 2140 810805
 .PNAME 63326 63376 92068-1X035 REV. 2101 800919
 .OPN7 63377 63422 24998-1X325 REV. 2101 800803
 .ERO.E 63423 63423 24998-1X249 REV. 2001 750701
 .FINH 63424 63703 24998-1X344 REV. 2140 810427
 SE.ZDQLB
 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.ZDRAGE
 SE.ZLTSB3
 DTMA 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.ZBULB
 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:ALBGS READY AT 10:36 AM MON., 17 SEPT, 1984

/LOADR:SEND

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

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, 7 1/2 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