MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1955-A
LOCAT - A DATA RETRIEVAL PROGRAM

by

Suzanne Y. Slinn
LOCAT - A DATA RETRIEVAL PROGRAM

by

Suzanne Y. Slinn
SARSAT Project Office
Electronics Division

DEFENCE RESEARCH ESTABLISHMENT OTTAWA
TECHNICAL NOTE 84-30

December 1984
Ottawa
ABSTRACT

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

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

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

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

RÉSUMÉ

Le présent 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'usager toute la souplesse voulue pour recouvrir les données dont il a besoin. Ainsi, celles-ci peuvent être rappelées d'après les paramètres suivants:

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

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

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

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

LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIGURE 1</td>
<td>LOCAT - Calling Sequence</td>
<td>2</td>
</tr>
<tr>
<td>FIGURE 2</td>
<td>SAT/LOC Soft Key Display</td>
<td>3</td>
</tr>
<tr>
<td>FIGURE 3</td>
<td>Frequency Soft Key Display</td>
<td>4</td>
</tr>
<tr>
<td>FIGURE 4</td>
<td>Output Soft Key Display</td>
<td>5</td>
</tr>
</tbody>
</table>
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/I00, 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.](image-url)
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.

![Frequency Soft Key Display](image-url)

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.](image)
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 "An output file is to be put on tape
IF (F704) I2 *4,4
F704:1 24
F604:1 24
ENDIF

ENDIF

C terminate the file selection with /E in the buffer
BUFF(1)=20/E
ENDIF

C send the transfer file command to be operating system
CALL EXEC(14,2,1001.1)
ENDIF

WRITE(14,1000)POSH,LOCLAT,LOCLNG,RADIUS,REGION,CASENO

C CONTINUE
CLOSE 'the database
CALL EXEC(15,1,1,1,ISTAT)
IF (ISTAT(1)) GO TO 999
WRITE(LIST,1005) ISTAT(1)

C CONTINUE
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 FORMAT

1000 FORMAT ("Enter target locataion data ")
1001 FORMAT ("Specify the search location")
1002 FORMAT ("Which data do you wish? ")
1003 FORMAT ("#ERROR# RADIUS specified incorrect")
1004 FORMAT ("IMAGE FILE ERROR (SELOC) ",15)
1005 FORMAT ("")
1006 FORMAT ("")
1007 FORMAT ("LOCATION LATITUDE - F10.4,2X,"LONCITUDE - ",
F10.4,2X,"RADIUS - ",F9.1)
1008 FORMAT ("Enter up to 5 specific satellites")
1009 FORMAT ("Enter up to 5 specific satellites")
1010 FORMAT ("Enter up to 5 specific satellites")
1011 FORMAT ("Do you wish another query?")
1012 FORMAT ("")
1013 FORMAT ("Enter SATPAS desired ie: CI 05407")
1014 FORMAT ("")
1015 FORMAT ("")
1016 FORMAT ("")
END
CLOSE(30)
CALL ALOCS
CALL EXEC(9,ANALOCs)
WRITE(10,101)ANSER
READ(LU,101)ANSER
OPEN(30,FILE='SCRATCH',STATUS='OLD')
READ(LU,101)PSN,START
CLOSE(30)
IF ((POSN - START) .NE. 0) FOUND = .TRUE.
IF (ANSWER .EQ. 2HY) GO TO 5
IF (FOUND) CALL OUP2
IF (FOUND) CALL OUTDT

store the output files onto tape if desired

IF (FOUND) THEN
IF (TAPE) THEN
IF (IMAGE) THEN
IF (SECOND) THEN
IF (THIRD) THEN
IF (FOURTH) THEN
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
clear the 'S' from the select array
DO 4 K=1, B
4 CONTINUE
C 210 CONTINUE
if specific satellites wanted, read the desired satellites
DO 114 I=1, 7
114 IF (SAT) THEN
WRITE(LU,1009)SPESAT(1),I=1,5
ENDIF
read desired lat,lon, radius if not all the locations are desired
IF (.NOT. (ALLUT)) THEN
WRITE (LU,1008)
CALL REUTL (LOCLAT, LOCLNG, FIG)
IF (FIG EQ 1) GO TO 999
CONTINUE
C 58 RADIUS = 10.
CALL CLONS (ANPRES, LU, SYS, I
- 124, 1,1, Radias, -12, IPARM)
IF (IPARM(1) ME 0) RADIUS * IPARM(1)
WRITE(1,554)
554 FORMAT (" ENTER THE REGION AND CASE NUMBER OF THIS LOCATION. ")
WRITE (1,557)
557 FORMAT (" THE CODES FOR THE REGIONS ARE: HZ - HALIFAX, /,
1 TO - TRENTO., ED - EDMONTON, T - VICTORIA ")
WRITE (1,556)
556 FORMAT (" ENTER THEN ACCORDING TO THIS EXAMPLE: TR 1111")
READ (1, 555)REGION, CASENO
555 FORMAT (A2, I4)
ENDIF
C determine if specific SATPAS is desired
IF (SPECIF) THEN
WRITE(LU,1013)
READ(LU,1014) (PASS(I), I=1, 4)
ENDIF
C determine which of the four outputs are desired
IF (POSN EQ 1) CALL OUTP
IF (POSN NE 1) THEN
OPEN (14, FILE='HEADR', IOSTAT=IOS, STATUS='OLD')
IF (IOS NE 1) WRITE (LU, 2619) IOS
IF (IOS NE 1) GO TO 999
219 FORMAT (' ERROR IN OPENING HEADR FILE ', J4)
DO 61 It,1.(PLACE-1)
READ(14, 1617)I
% FORMAT (A2)
CONTINUE
ENDIF
C IF (EXIT) GO TO 999
C determine which of the frequencies are desired
CALL FREQ
C GO TO 999
C display lat, lon, radius
WRITE(14, 1810) PSNM, LOCLAT, LOCLNG, RADIUS, REGION, CASENO
PLACE+PLACE+1
START : POSN
CLOSE(14)
C IF (IFRM(AN) LT 8) GO TO 999
C Write the common variables to a scratch file
OPEN(30 FILE='SCRATCH', STATUS='OLD')
WRITE(30,1010) ALLUT, ALT, SAT, S1, S2, EX, (SPESAT(I), I=1, 5), RNM,
SECOND, THIRD, FOURTH
1010 FORMAT (5(I5, 12), SA2, IX, 4L1, IX)
WRITE(30,1019) PSNM, S1, S2, UNIT, BIT, BENT, REALT, COS, CRNC12,
C BIC12, ALLUT, POSN, START, EXIT, IPC, SPEC, SEAVENT,
1 PASS(I), I=1, 4
1019 FORMAT (5(I5,12), F10.3, IX, 4L1, IX) 2(F5.3, IX, 4L1, IX), 15, 4A2)
WRITE(30,1020) (LOCLAT(I), I=1, 4), (LOCLNG(I), I=1, 4), (REGION(I), I=1, 4),
1 (BOUND(I), I=1, 5), RANGE, LOCLAT, LOCLNG, RADIUS, IMAGE, REGION,
1 CASENO, PLACE, (IMAGE(I), I=1, 5)
1020 FORMAT (5(A2, IX, 14, IX, 13, 4A2)
1 L1, A2, IX, 14, IX, 13, 4A2)
EXECUTABLE CODE

DO 1 I=1,5
     READ(UNIT,*)IBASE(I)
     IF (IBASE(I).NE.1) GO TO 1
     I = I + 1
     IF (I.EQ.5) GO TO 1

WRITE(LU,*)IBASE(1:11)
READ(LU,*)ISYS
IF (ISYS.EQ.0) GO TO 3
IF (ISYS.EQ.1) GO TO 4
FOUN = .FALSE.
EXIT = .FALSE.
TAPE = .FALSE.
REGION = 2
CASENO = 0
PI = 1

initialize variables

CONTINUE

ALLUT = .TRUE.
ALSAT = .TRUE.
SAT = .FALSE.
CX = .FALSE.
SX = .FALSE.
SPECIF = .FALSE.
UNIT = 99
LU = .FALSE.
MC = .FALSE.
SAR = .TRUE.
TEST = .FALSE.
ALL = .FALSE.
LOCLAT = .FALSE.
LOCNC = .FALSE.
RADIUS = .FALSE.

set other input variables

CALL SESET(UNIT,RET)
IF (RET.NE.1) GO TO 99

search files specified

CONTINUE

CALL FUNKY(KEYS,PROM)
WRITE(113,103)
CALL RE10(LU,RK,-2)
UK = RK -24f0
IF (RK.EQ.1) SAT = .TRUE.
IF (RK.EQ.2) ALLUT = .FALSE.
IF (RK.EQ.3) CX = .TRUE.
IF (RK.EQ.4) SX = .TRUE.
IF (RK.EQ.5) SPECIF = .TRUE.
GO TO 182
ENDIF

IF (KEYS(I,RK).EQ.2) THEN
    KEYS(I,RK) = 2
    IF (KEYS(I,RK).EQ.1) SAT = .TRUE.
    IF (KEYS(I,RK).EQ.2) ALLUT = .FALSE.
    IF (KEYS(I,RK).EQ.3) CX = .TRUE.
    IF (KEYS(I,RK).EQ.4) SX = .TRUE.
    IF (KEYS(I,RK).EQ.5) SPECIF = .TRUE.
    GO TO 182
ENDIF

IF (KEYS(I,RK).EQ.3) THEN
    KEYS(I,RK) = 3
    IF (KEYS(I,RK).EQ.1) SAT = .TRUE.
    IF (KEYS(I,RK).EQ.2) ALLUT = .FALSE.
    IF (KEYS(I,RK).EQ.3) CX = .TRUE.
    IF (KEYS(I,RK).EQ.4) SX = .TRUE.
    IF (KEYS(I,RK).EQ.5) SPECIF = .FALSE.
ENDIF

IF ((NOT(SAT AND CX AND SX))ALSAT = .TRUE.

CALL FUNKY(KEYS(I,RK),RK)
IF (RK.EQ.6) GO TO 200
GO TO 181

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 IMODE(4), ILEVEL(3), IMODE, ISTAT(28), LIST
INTEGER IUTATE(3), IUTST(3), IUTAR(3), ITMCC(3), ITSPD(3)
INTEGER SUFB(47), SUFF(47), IMUBF(76), FRUB(58)

COMMON /BASE/ IMODE, ILEVEL, IMODE, ISTAT, LIST, ITATE,
- IUTATE, IUTST, IUTAR, ITMCC, ITSPD, SUFF, IMUBF, SUFF
- IMUBF, SUFF, IMUBF, FRUB, FRUB

DOUBLE PRECISION QTINE, SDT

- QTINE - start date
- SDT - start date

C - latitude of specified location
C - longitude of specified location
C - maximum distance between specified location and file locations

INTEGER IBASE(3), ILEVEL(3), IMODE, ISTAT(28), LIST
INTEGER IUTATE(3), IUTST(3), IUTAR(3), ITMCC(3), ITSPD(3)
INTEGER SUFB(47), SUFF(47), IMUBF(76), FRUB(58)

INTEGER LU, RET
INTEGER SYL, SYST, LOGLU

REAL ONEDAY

INTEGER KEYS(9)

INTEGER FLG, AA, IFBRK

DATA KEYS(9) = '1', '2', '3', '4', '5', '6', '7', '8', '9'

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

COMMON/SELECT/ALLUT, ALSAT, SX, CX, SPESAT, PRIM, SECOND, THIRD,
- FOURTH, PST, STS, PST, UNIT, SST, PENT, REALT, CD04, CDIC12,
- CDRC12, ALLUT, PAND, START, EXIT, TAPE, SPECIES, SEVENT, PASS,
- UNIT, DENT, USER, TYPE, ELTANG

INTEGER CONTRY, USER, TYPE, ELTANG

COMMON /DENT/ ALLUT, ALSAT, SX, CX, PRIM, SECOND, THIRD, FOURTH, SPECIES,
- REALT, CD04, CDIC12, CDRC12, ALLUT, PAND, START, EXIT, TAPE

REAL CTANG

INTEGER SPESAT(15), PST, SST, PST, FILE, START, ANSWER

INTEGER IMUBF(76), SEVENT, PASS(47), REGION, CAUSE

INTEGER FILE

DATA KEYS // Specific SAT A location, SAT, CX, SPESAT, PRIM, SECOND, THIRD,
- FOURTH, PST, SST, PST, UNIT, SST, PENT, REALT, CD04, CDIC12,
- CDRC12, ALLUT, PAND, START, EXIT, TAPE

DATA PRIM // '1123456789' /

C DATA KEYS // Array containing contents of soft keys
C PRIM // Array of values to return for soft keys
C RX // actual value returned

DATA IBASE// TR, ST, PRIM, */
DATA FILEVL /6HENTRY /
DATA LIST /6HGROUP / 
DATA IDLE /5HFIELD / 
DATA ITDTE /6HDATE /
DATA ITTST /6HTEST /
DATA ITSAR /6HSARN/
DATA ITNCC /&MCCREF/
DATA ITSPD /5HSATPAS/
DATA IDTST /6HFIELD /
DATA IDSAR /6HSARIF /
DATA IDNCC /6HNCCELF/
DATA IDLUT /6HLUTELF/
END

FILN4

SFILES(IS,15)

PROGRAM LOCAT (3 99)
SET-48-88 V00 (040726.9987)

IMPLICIT NONE

C

C SARSAT EVALUATION FACILITY

C

C DATE VERSION DESCRIPTION AUTHOR

C 83/10/85 00 -- -- -- SUZANNE Y. SLINN

C

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:

ENPAR

PARSE

IN

User:

MOVES - move a byte string
SESET - to copy the operator for the date range, the device for the output file
FUNKY - to set soft keys (ISCRN in ISEFIL)
RETL - copy operator for latitude and longitude
OPENH - to open the database
CLOSE - to close the database

Functions:

GETCH - to get a character from a string
NUM2INT - to convert ASCII string to integer
FTIME - to convert integer times to double precision seconds from 1990

C

DATA DECLARATIONS

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

INTEGER LODAT, HIDAT, OUTLU, RANGE

LODAT - start date in ASCII TMDMD
HIDAT - end date in ASCII TMDMD
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
EN DAT - end date in double precision seconds from 1980

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

REAL LOCAT, LOCLNG, RADIUS

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

COMMON /LOCN/ LOCAT, LOCLNG, RADIUS

INTEGER IBASE(5), ILVL(3), IMODE, ISTAT(18), LIST

INTEGER ITDTE(3), ITTST(3), ITTST(3), ITMC(3), ITMPD(3)

INTEGER IBDL(3), ITDST(3), IBDL(3), IMDC(3), IDMC(3)

these are the parameters required by the IMAGE subroutines

ITDTE, ITTST, IBDL, IMDC are the parameters for each of the files accessed
IDMC, IMDC are the ID parameters for each of the files

INTEGER SBUF(68), LBUF(99), MBUF(76), FBUF(5)

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

COMMON /BASE/ IBASE, ILVL, IMODE, ISTAT, LIST, ITDTE,
- ITTST, IBDL, IMDC, ITMPD, IDMC, -
- IBDL, IDST, SBUF, LBUF, MBUF, FBUF

COMMON /SELECT/ ALLUT, ALSAT, SAT, SX, CX, SPESAT, PRIM, SECOND, THIRD,
- FOURTH, PST, SAT, TST, UNIT, D1F, BENT, REAL, COS4, CMC24,
- CDC24, ALFSAT, PSTR, START, EXIT, TAPE, SPECIF, SEVENT, PASS,
- SPECIF, SARSAT, EXIT, TAPE, SPECIF, SEVENT, PASS

INTEGER CMC24, ALLFSAT, PSTR, START, EXIT, TAPE, SPECIF, SEVENT, PASS

REAL DIFF

REAL DIFF

REAL DIFF

REAL DIFF

REAL DIFF

REAL DIFF

REAL DIFF
APPENDIX B

SOURCE CODE LISTINGS
<table>
<thead>
<tr>
<th>THIRD OUTPUT</th>
<th>PRIMARY LOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRIFT</td>
<td>CTAI</td>
</tr>
<tr>
<td>1)</td>
<td>0 0000</td>
</tr>
<tr>
<td>2)</td>
<td>0 0000</td>
</tr>
<tr>
<td>3)</td>
<td>0 0111</td>
</tr>
<tr>
<td>4)</td>
<td>0 0000</td>
</tr>
<tr>
<td>5)</td>
<td>0 0000</td>
</tr>
<tr>
<td>6)</td>
<td>0 0000</td>
</tr>
<tr>
<td>7)</td>
<td>0 0000</td>
</tr>
<tr>
<td>8)</td>
<td>1 5644</td>
</tr>
<tr>
<td>9)</td>
<td>3 5311</td>
</tr>
<tr>
<td>10)</td>
<td>0 0000</td>
</tr>
<tr>
<td>11)</td>
<td>0 0000</td>
</tr>
<tr>
<td>12)</td>
<td>0 0000</td>
</tr>
<tr>
<td>13)</td>
<td>0 0000</td>
</tr>
<tr>
<td>14)</td>
<td>7 3387</td>
</tr>
<tr>
<td>15)</td>
<td>0 0000</td>
</tr>
<tr>
<td>16)</td>
<td>0 0189</td>
</tr>
<tr>
<td>17)</td>
<td>0 0000</td>
</tr>
<tr>
<td>18)</td>
<td>0 0000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FOURTH OUTPUT</th>
<th>PRIMARY LOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1VCTA</td>
<td>V1VICTA</td>
</tr>
<tr>
<td>1)</td>
<td>0 1533</td>
</tr>
<tr>
<td>2)</td>
<td>0 1138</td>
</tr>
<tr>
<td>3)</td>
<td>0 8235</td>
</tr>
<tr>
<td>4)</td>
<td>0 8971</td>
</tr>
<tr>
<td>5)</td>
<td>0 8776</td>
</tr>
<tr>
<td>6)</td>
<td>0 8278</td>
</tr>
<tr>
<td>7)</td>
<td>0 1087</td>
</tr>
<tr>
<td>8)</td>
<td>0 9566</td>
</tr>
<tr>
<td>9)</td>
<td>0 9422</td>
</tr>
<tr>
<td>10)</td>
<td>0 9818</td>
</tr>
<tr>
<td>11)</td>
<td>0 9478</td>
</tr>
<tr>
<td>12)</td>
<td>0 6158</td>
</tr>
<tr>
<td>13)</td>
<td>0 8618</td>
</tr>
<tr>
<td>14)</td>
<td>0 8486</td>
</tr>
<tr>
<td>15)</td>
<td>0 8589</td>
</tr>
<tr>
<td>16)</td>
<td>0 1784</td>
</tr>
<tr>
<td>17)</td>
<td>0 8745</td>
</tr>
<tr>
<td>18)</td>
<td>0 8689</td>
</tr>
<tr>
<td>PRIMARY DATA</td>
<td>SECONDARY DATA</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------</td>
</tr>
<tr>
<td>DATE SATPAS</td>
<td>LOCATION LATITUDE</td>
</tr>
<tr>
<td>D1</td>
<td>04/66</td>
</tr>
<tr>
<td>D2</td>
<td>04/66</td>
</tr>
<tr>
<td>D3</td>
<td>04/66</td>
</tr>
<tr>
<td>D4</td>
<td>04/66</td>
</tr>
<tr>
<td>D5</td>
<td>04/66</td>
</tr>
<tr>
<td>D6</td>
<td>04/66</td>
</tr>
</tbody>
</table>

SECOND OUTPUT PRIMARY LOCATION

<table>
<thead>
<tr>
<th>CTA</th>
<th>POINTS</th>
<th>TREND</th>
<th>QUAL</th>
<th>PROB</th>
<th>MNWLS</th>
<th>TCA</th>
<th>QTIME</th>
<th>LOSTFIN</th>
<th>DIAS</th>
<th>CORR</th>
<th>SCORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-19 4215</td>
<td>52 36 4722</td>
<td>24 6199</td>
<td>137 49</td>
<td>4 5 2318</td>
<td>5 1881</td>
<td>2 7714</td>
<td>1244 2024 10474 14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>-19 4215</td>
<td>52 36 4722</td>
<td>24 6199</td>
<td>137 49</td>
<td>4 5 2318</td>
<td>5 1881</td>
<td>2 7714</td>
<td>1244 2024 10474 14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-19 4215</td>
<td>52 36 4722</td>
<td>24 6199</td>
<td>137 49</td>
<td>4 5 2318</td>
<td>5 1881</td>
<td>2 7714</td>
<td>1244 2024 10474 14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>-19 4215</td>
<td>52 36 4722</td>
<td>24 6199</td>
<td>137 49</td>
<td>4 5 2318</td>
<td>5 1881</td>
<td>2 7714</td>
<td>1244 2024 10474 14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-19 4215</td>
<td>52 36 4722</td>
<td>24 6199</td>
<td>137 49</td>
<td>4 5 2318</td>
<td>5 1881</td>
<td>2 7714</td>
<td>1244 2024 10474 14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>-19 4215</td>
<td>52 36 4722</td>
<td>24 6199</td>
<td>137 49</td>
<td>4 5 2318</td>
<td>5 1881</td>
<td>2 7714</td>
<td>1244 2024 10474 14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>-19 4215</td>
<td>52 36 4722</td>
<td>24 6199</td>
<td>137 49</td>
<td>4 5 2318</td>
<td>5 1881</td>
<td>2 7714</td>
<td>1244 2024 10474 14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>-19 4215</td>
<td>52 36 4722</td>
<td>24 6199</td>
<td>137 49</td>
<td>4 5 2318</td>
<td>5 1881</td>
<td>2 7714</td>
<td>1244 2024 10474 14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>-19 4215</td>
<td>52 36 4722</td>
<td>24 6199</td>
<td>137 49</td>
<td>4 5 2318</td>
<td>5 1881</td>
<td>2 7714</td>
<td>1244 2024 10474 14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>-19 4215</td>
<td>52 36 4722</td>
<td>24 6199</td>
<td>137 49</td>
<td>4 5 2318</td>
<td>5 1881</td>
<td>2 7714</td>
<td>1244 2024 10474 14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>-19 4215</td>
<td>52 36 4722</td>
<td>24 6199</td>
<td>137 49</td>
<td>4 5 2318</td>
<td>5 1881</td>
<td>2 7714</td>
<td>1244 2024 10474 14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>-19 4215</td>
<td>52 36 4722</td>
<td>24 6199</td>
<td>137 49</td>
<td>4 5 2318</td>
<td>5 1881</td>
<td>2 7714</td>
<td>1244 2024 10474 14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>-19 4215</td>
<td>52 36 4722</td>
<td>24 6199</td>
<td>137 49</td>
<td>4 5 2318</td>
<td>5 1881</td>
<td>2 7714</td>
<td>1244 2024 10474 14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>-19 4215</td>
<td>52 36 4722</td>
<td>24 6199</td>
<td>137 49</td>
<td>4 5 2318</td>
<td>5 1881</td>
<td>2 7714</td>
<td>1244 2024 10474 14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>-19 4215</td>
<td>52 36 4722</td>
<td>24 6199</td>
<td>137 49</td>
<td>4 5 2318</td>
<td>5 1881</td>
<td>2 7714</td>
<td>1244 2024 10474 14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>-19 4215</td>
<td>52 36 4722</td>
<td>24 6199</td>
<td>137 49</td>
<td>4 5 2318</td>
<td>5 1881</td>
<td>2 7714</td>
<td>1244 2024 10474 14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>-19 4215</td>
<td>52 36 4722</td>
<td>24 6199</td>
<td>137 49</td>
<td>4 5 2318</td>
<td>5 1881</td>
<td>2 7714</td>
<td>1244 2024 10474 14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>-19 4215</td>
<td>52 36 4722</td>
<td>24 6199</td>
<td>137 49</td>
<td>4 5 2318</td>
<td>5 1881</td>
<td>2 7714</td>
<td>1244 2024 10474 14</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX A
SAMPLE RUN

LOCAT
Enter the name of the database to be used
SAR2-MW
Enter date for start and end of search; i.e. YYMMDD YYMMDD
840618 840618
Specify retrieval output device LU: 18

Specific SAT & A location Cx Sx
Satpass # END of select EXIT

Enter target location data:
Specify the search location
Longitude: -79.8833
Latitude: 45.35
Radius: 400

ENTER THE REGION AND CASE NUMBER OF THIS LOCATION.
THE CODES FOR THE REGIONS ARE: "HZ" - HALIFAX,
"TR" - TRENTON, "ED" - EDMONTON, "YJ" - VICTORIA.
ENTER THEN 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 frequencies EXIT

Which output do you wish?

406 Bent Pipe 406 Real Time 406 Cospas # CBC121
# CBC243 # END of select All frequencies EXIT

Which output do you wish?

There are 16 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 ;ALOC5 have been created to load the main program and its related subroutines. To load the programs, enter:

:RU,LOADR,;LOCAT
:RU,LOADR,;ALOC5

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,ALOC5::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:

YYMDD YYMDD

• 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.
SUBROUTINE OUTPT
         IMPLICIT NONE

COMMON /LOGC/ LODAT(3), HIDAT(3), SIDAT, ENDAT, OUTLU(5), RANGE
INTEGER LODAT, HIDAT, OUTLU, RANGE
DOUBLE PRECISION SIDAT, ENDAT

REAL LOCLAT, LOCLNG, RADIUS

COMMON /LOCN/ LOCLAT, LOCLNG, RADIUS

INTEGER IBASE(5), ILEVEL(3), IMODE, ISTAT(9), LIST
INTEGER ITDATE(3), ITTST(3), ITSAR(3), ITMCC(3)
INTEGER IDLUT(3), IDTST(3), IDSAR(3), IDMCC(3)
INTEGER IPARM(5), PDUF(33)
INTEGER LU, RET
INTEGER SYS, ISYS, LOGLU

LOGICAL LUT, MCC, SAR, TEST, ALL

COMMON /BASE/ IBASE, ILEVEL, IMODE, ISTAT, LIST, ITDATE,
             ITTST, ITSAR, ITMCC, IDLUT, IDTST, IDSAR, IDMCC,
             IDLUT, IDTST, PDUF, UNIX, MBUF, FBUF

DOUBLE PRECISION QTME, SDT

SOT - start date

LODAT - ascii start date YYMMDD
SIDAT - low date in seconds from 1980
ENDAT - high date in seconds from 1980
OUTLU - output device, LU or filename

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

RBUF - receiving buffer
IPARM - return buffer for RMPAR call
FBUF - 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 SAR STAT incident detail file selected
TEST - flag to indicate FIELD test master file selected
ALL - all files will be searched
INTEGER KEYS(4), PROM(8), RK
INTEGER FLG, AA, IFIBR

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

COMMON/SELECT/ALLOT, ALSAT, SAT, SX, CX, SPESAT, PRIM, SECOND, THIRD,
   PRIM, SAT, SST, TST, PST, UNIT, UF, RET, REAL, COS4, CRC12,
   CRC24, ALLFR, POSM, START, EXIT, TAPE, SPECIF, SEVENT, PASS, IMAGE,
   CONTRY, USER, TYPE(4)
INTEGER CONTRY(3), USER, TYPE(4)
REAL DIF

**EXECUTABLE CODE**

ISYS = LOGLUA(SYS)
initialize variables
LU = 1
UNIT = 99
search files specified
CONTINUE

CALL FUNKY(KEYS, PROM)
WRITE (LU, 183)
CALL READ1(LU, RK, 2)
RK = RK - 246
IF (RK LT 1 OR RK GT 8) GO TO 101
IF (RK EQ 8) THEN
   EXIT = .TRUE.
   GO TO 999
ENDIF

CALL FUNKY(KEYS(LU, RK, RK))
IF (RK EQ 6) GO TO 208
GO TO 111
211 CONTINUE

CALL FUNKY(KEYS(LU, RK))
IF (RK EQ 6) GO TO 208
GO TO 181

CONTINUE
change '0' to ' ' DO 4 K=1,9 KEYS(K,KE)=24 CONTINUE

C open the header file
OPEN(14,FILE='HEADR',IOMAT=10S,STATUS='OLD')
IF (IOM .NE. 1) WRITE(LU,1886)10S
IF (IOM .NE. 0) GO TO 999

999 CONTINUE
WRITE(L1U,1005)
1001 FORMAT( ERROR IN OPENING PRIMARY. IOS IS ,14)
1002 FORMAT( ERROR IN OPENING SECOND. IOS IS ,14)
1003 FORMAT( WHICH OUTPUT DO YOU WISH ? )
1004 FORMAT( ERROR IN OPENING FOURTH. IOS IS ,14)
1005 FORMAT( ERROR IN OPENING HEADR. IOS IS ,14)
RETURN
END

FTN4
$FILES(iS is)
SUBROUTINE OUTP2(PLACE)
,SEF-400
V86
(84172S.1426)
IMPLICIT NONE
C $CARSAT EVALUATION FACILITY $C ..--..----.-.-.------- $C SARSAT EVALUATION FACILITY $C ..--..----.-.-.------- $C DATE VERSION DESCRIPTION AUTHOR
83/10/05 00 ----- SUZANNE Y. SLINN
C DESCRIPTION:
This subroutine opens the output files.
C CALLING SEQUENCE:
C CALLED FROM: LOCAT
C CALL OUTPT
C DATA DECLARATIONS
COMMON /LOGC/ LODAT(3), HIDAT(3), SFORMAT, EFORMAT, QOUTLU(5), RANGE
INTEGER PLACE
INTEGER LODAT, HIDAT, OUTLU, RANGE
DOUBLE PRECISION SFORMAT, EFORMAT
REAL LOCLAT, LOCLNG, RADIUS
C LOCLAT - latitude of specified location
C LOCLNG - longitude of specified location
C RADIUS - maximum distance between specified location and
C file locations
COMMON /LOCN/ LOCLAT, LOCLNG, RADIUS
C INTEGER IBASE(5), ILEVEL(3), IMODE, IFORMAT(10), LIST
C INTEGER IDMCC(3), IDTST(3), IDSAR(3), ITSPD(3)
C INTEGER IDLUT(3), IDTST(3), IDSAR(3), IDMCC(3)
INTEGER SBUF(68), LDUF(9), MBUF(76), FIUF(56)
COMMON /BASE/, IBASE, ILEVEL, IMODE, ISTAT, LSTAT, ITODE, 
   IISTAT, ITJAB, JTMC, ITSPD, IOUT, ITMCC,
   IDVAR, IBSTAT, SBUF, LDUF, MBUF, FIUF

DOUBLE PRECISION QTIME, SDT

SDT - start date

LODAT - ascii start date YYMMDD
STAT - low date in seconds from 1980
RENT - high date in seconds from 1980
OUTL - output device, LU or filename

INTEGER RBUF(4), IPARM(5), RBUF(33)
INTEGER LSYS, LSTAT, LOGLU

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

REAL ONEFDAY
INTEGER DATCH(3), LOOP, DUN(7), II

INTEGER UNIT

LOGICAL LUT, SAC, TST, 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
TST - flag to indicate TST master file selected
ALL - all files will be searched

INTEGER KEYS(8, B), PROM(6), RK

INTEGER FLC, MA, IPARX

FLG - flag to indicate if FLGX used
MA - dummy parameter for IPARX

COMMON /SECLT/, ALLUT, LUT, SAT, SX, CX, SPESAT, PRIM, SECOND, THIRD,
   FOURTH, PST, SST, TST, TST, UNIT, DIF, BENT, REALT, COS4, CAC12,
   CAC24, ALLFR, POSN, ST, EXIT, TAP, SPECIF, SEVENT, PASS, IMAGE,
   CONTR, USER, TYPE, ELTANG

INTEGER CONTY(3), LUSER, TYPE(4)

REAL ELTANG

LOGICAL ALLUT, ALSAT, SX, CX, PRIM, SECOND, THIRD, FOURTH

LOGICAL BENT, REALT, COS4, CAC12, CAC24, ALLFR, EXIT, TAP, SPECIF

LOGICAL IMAGE

INTEGER SPESAT(5), TOS, PST, SST, TST, 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
SX - flag to indicate if specific satellites wanted
SPESAT - flag to indicate if all SARSAT satellites wanted
SPESAT - array that contains the specific satellites desired

EXECUTABLE CODE

IST = LOGLU(ISTS)

initialize variables

LU = 1
UNIT = 99

search files specified

CONTINUE

OPEN(I4, FILE='HEADR', IOSTAT=10, STATUS='OLD')

IF (10 EQ. 0) WRITE(LU, (1000)) 101
IF (10 EQ. 0) GO TO 100
open primary file if this output is desired

IF (PRIM) THEN
  OPEN(I6,FILE='PRIMARY',IOSTAT=IOS,STATUS='OLD')
  IF (IOS .NE. 0) WRITE(LU,1001)IOS
ENDIF

open second file if this output is desired

IF (SECOND) THEN
  OPEN(I7,FILE='SECOND',IOSTAT=IOS,STATUS='OLD')
  IF (IOS .NE. 0) WRITE(LU,1002)IOS
ENDIF

open third file if this output is desired

IF (THIRD) THEN
  OPEN(I8,FILE='THIRD',IOSTAT=IOS,STATUS='OLD')
  IF (IOS .NE. 0) WRITE(LU,1003)IOS
ENDIF

open fourth file if this output is desired

IF (FOURTH) THEN
  OPEN(I9,FILE='FOURTH',IOSTAT=IOS,STATUS='OLD')
  IF (IOS .NE. 0) WRITE(LU,1004)IOS
ENDIF

IF (IMAGE) THEN
  IF (PRIM) THEN
    OPEN(21,FILE='PRIMARY',IOSTAT=IOS,STATUS='OLD')
    IF (IOS .NE. 0) WRITE(LU,1011)IOS
    IF (IOS .NE. 0) GO TO 990
  ENDIF
  IF (SECOND) THEN
    OPEN(22,FILE='SECOND',IOSTAT=IOS,STATUS='OLD')
    IF (IOS .NE. 0) WRITE(LU,1012)IOS
    IF (IOS .NE. 0) GO TO 990
  ENDIF
  IF (THIRD) THEN
    OPEN(23,FILE='THIRD',IOSTAT=IOS,STATUS='OLD')
    IF (IOS .NE. 0) WRITE(LU,1013)IOS
    IF (IOS .NE. 0) GO TO 990
  ENDIF
  IF (FOURTH) THEN
    OPEN(24,FILE='FOURTH',IOSTAT=IOS,STATUS='OLD')
    IF (IOS .NE. 0) WRITE(LU,1014)IOS
    IF (IOS .NE. 0) GO TO 990
  ENDIF
ENDIF

END IF

CONTINUE

IF (PRIM) THEN
  READ(10,2017)
IF (SECOND) READ(11,2017)
IF (THIRD) READ(2,2017)
IF (FOURTH) READ(3,2017)
IF (IMAGE) THEN
  IF (PRIM) READ(21,2017)
  IF (SECOND) READ(22,2017)
  IF (THIRD) READ(23,2017)
  IF (FOURTH) READ(24,2017)
ENDIF

FORMAT(2017)
65 CONTINUE
DO %I 11=1,PLACE-1)
  READ(14,2817)
96 CONTINUE
ENDIF

WRITE(LU,1005)
1000 FORMAT(' ERROR IN OPENING PRIMARY. IOS IS ', IOS)
1001 FORMAT(' ERROR IN OPENING SECOND. IOS IS ', IOS)
1002 FORMAT(' ERROR IN OPENING THIRD. IOS IS ', IOS)
1003 FORMAT(' Which output do you wish?')
1004 FORMAT(' ERROR IN OPENING FOURTH. IOS IS ', IOS)
1005 FORMAT('
RETURN
END
SUBROUTINE FREQ
IMPLICIT NONE
C
SARSAT EVALUATION FACILITY
C
DATE VERSION DESCRIPTION AUTHOR
03/18/85 88 -------------- SUZANNE Y. SLINN
C
DESCRIPTION:
This subroutine determines which of the frequencies the user desires to use.
CALLING SEQUENCE:
CALLED FROM: LOCAT
CALL FREQ
C
DATA DECLARATIONS
C
COMMON /LOC/ LODAT(3), MIDAT(3), STDAT, ENDAT, OUTFI(5), RANGE
INTEGER LODAT, MIDAT, OUTFI, RANGE
DOUBLE PRECISION STDAT, ENDAT
REAL LOCAT, LOCIMG, RADIUS
LOCAT - latitude of specified location
LOCIMG - longitude of specified location
RADIUS - maximum distance between specified location and file locations
COMMON /LOC/ LOCAT, LOCIMG, RADIUS
C
INTEGER IBASE(5), ILEVEL(3), IMODE, ISTAT(10), LIST
INTEGER ITDIST(3), ITSAR(3), ITIMEC(3), ITSPD(3)
C
INTEGER ILODUT(31), IDLUT(3), IDSAR(3), IDNCC(3)
 INTEGER SBUF(69), L3UF(99), MBUF(76), FBUF(59)
 COMMON /BASE/ IBASE, ILEVEL, IMODE, ISTAT, LIST, ITDIST,
 - ITIMEC, ITSAR, ITSPD, ILODUT, IDLUT, IDNCC,
 - IDMCH, IDIST, SBUF, L3UF, MBUF, FBUF
DOUBLE PRECISION QTIME, SDT
SDT - start date
LODAT - asci start date YYMMDD
STDAT - low date in seconds from 1991
ENDAT - high date in seconds from 1991
OUTLI - output device, LU or Filename
INTEGER PRUF(49), IPARM(5), PRUF(33)
INTEGER LU, RET, SYSC, ISYS, LOGLU
PRUF - receiving buffer
IPARM - return buffer for RRPAR call
PRUF - return buffer from PARSE
REAL ONEWAY
INTEGER MATCH(3), LOOP, DUN(7), IL
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), RE
INTEGER FLG, AA, IFRK

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

COMMON/SELECT/ALL, LUT, MCC, ISO, SX, SAL, SECOND, THIRD,
FIRST, PST, SST, TST, UNIT, DIF, BENT, REALT, CODE,
COS4, CBC12, CBC24, ALLFR, POSN, START, EXIT, TAPE, SPECIF, SEV, PASS,
DIF, CTRM, DIRT, JUGER, TYPE(14)

LOGICAL IMAGE

LOGICAL ALL, LUT, MCC, ISO, SX, FIRST, SECOND, THIRD, FOURTH
LOGICAL BENT, REALT, CODE, CBC12, CBC24, ALLFR, EXIT, TAPE, SPECIF

INTEGER SEV, PASS(4)

REAL DIF

ALLUT - flag to indicate if all locations wanted
ALLSAT - flag to indicate if all satellites wanted
SAT - flag to indicate if specific satellites wanted
ALL - flag to indicate if all SARSTAT satellites wanted
SAR - flag to indicate if all SARSTAT satellites wanted
SPESAT - array that contains the specific satellites desired

DATA KEYS /* 486 Bent Pipe 486 Real Time */
- /* 486 Cospas CBC12 */
- /* 486 Cospas END of select */
- /* All frequencies EXIT */

DATA PROM /* 51/256/59/7/9 */

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

*********************************************************************************************

EXECUTABLE CODE

*******************************************************************************

ISYS = LOGLU(SYS)

initialize variables
ALLFR = FALSE
BENT = FALSE
REALT = FALSE
CODE = FALSE
CBC12 = FALSE
CBC24 = FALSE
LU = 1
UNIT = 99

search files specified

100 CONTINUE

CALL FUNKY (KEYS, PROM)
WRITE (15,1003)

101 CALL READ (LU, RX,-2)
RX = RX - 9948
IF (RX LT 1 OR RX GT 8) GO TO 101
IF (RX EQ 0) THEN
EXIT = TRUE.
GO TO 999
ENDIF

set R in key selected and check for further selects

IF (KEYS(RX) EQ 29) THEN
KEYS(RX) = 298
IF (RX EQ 1) BENT = TRUE
IF (RX EQ 2) REALT = TRUE.
IF (RX EQ 3) CODE = TRUE
IF (RX EQ 4) CBC12 = TRUE
IF (RX EQ 5) CBC24 = TRUE
IF (RX EQ 7) ALLFR = TRUE.
GO TO 102
ENDIF

if character
IF (KEYS(1,RK) EQ. 2H) THEN  
   KEYS(1,RK) = 2H  
   IF (RK EQ. 1) THEN DENT = .FALSE.  
   IF (RK EQ. 2) THEN REALT = .FALSE.  
   IF (RK EQ. 3) THEN COS4 = .FALSE.  
   IF (RK EQ. 4) THEN CBC24 = .FALSE.  
   IF (RK EQ. 5) THEN ALLFR = .TRUE.  
ENDIF

182 CALL FUNKY(KEYS(1,RK),RK)  
   IF (RK EQ. 6) GO TO 200
   GO TO 181

200 CONTINUE
   clear the '6' from the select array
   DO 4 RK=1,8  
      KEYS(1,RK) = 2H
   CONTINUE

999 CONTINUE
   WRITE(LU1,1885)  
   1885 FORMAT('ERROR IN OPENING PRIMARY. IOS IS ',14)  
   1886 FORMAT('ERROR IN OPENING SECOND. IOS IS ',14)  
   1887 FORMAT('ERROR IN OPENING THIRD. IOS IS ',14)  
   1888 FORMAT('ERROR IN OPENING FOURTH. IOS IS ',14)  
   1889 FORMAT('')  
   RETURN
END

FILE(15,15)  
SUBROUTINE OUTD(ICONTR)  
   DATE  VERSION  DESCRIPTION  AUTHOR  
   03/11/85  #0  -------  SUZANNE Y. SLINN

   COMMON /LOC/ LODAT(3), HDAT(3), SDATE, ENDATE, OUTLU(5), RANGE  
      INTEGER LODAT, HDAT, OUTLU, RANGE  
      DOUBLE PRECISION SDATE, ENDATE  
   REAL LOCLAT, LOCLNG, RADIUS, LUTLA, LUTLG, DSTNC  
   LOCLAT - latitude of specified location  
   LOCLNG - longitude of specified location  
   RADIUS - radius distance between specified location and  
            file locations
COMMON /LOC/ LOCLAT, LOCLNG, RADIUS

INTEGER IBASE(3), ILEV(3), IMODE, ISTAT(3), LIST
INTEGER IDTCT(3), IDTST(3), ITSAR(3), ITMCC(3), ITSPD(3)
INTEGER IBUF(40), IPARM(5), PBUF(33)
INTEGER UNIT
INTEGER DATCH(3), LOOP, LBUF, NBUF, FBUF

DOUBLE PRECISION GTIME, SDT

SDT = start date

LDAT = earliest start date YYMMDD
STDAT = latest start date in seconds from 1980
ENDAT = highest date in seconds from 1980
OUTLUT = output device, LU or filename

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

DOUBLE PRECISION QTINE, SOT

CDT = start date
LCUT = start date in seconds from 1990
ENDT = end date in seconds from 1990
OUTLUT = output device, LU or filename

INTEGER KEYS(8), PROM(9), RK

INTEGER FLC, FLG, A.

REAL DIVER, DIF, R, P

REAL ELTANG

LOGICAL ALLUTALSAT SAT SX CX, PRIM SECOND, THIRD, FOURTH

LOGICAL RENT

LOGICAL ALLFR, EXIT, TAPE, SPECIF

LOGICAL ALLUT

INTEGER SPESAT(S), IPST, SST, TST, MSG1(R), POSN, START

REAL SEVENT, PASS(4)

REAL LATC, LONGT, LAT, LONGCFA, DIFR, DLAT, DLONG

INTEGER TRYA(6), TRYB(6), LINE, PAGE, ESRC, MLINE, POS, CP0S

INTEGER MPASS(6), MEVET, PASS(4), LEVEL, ITEM(3), MREF, MSG2(60)

REAL LONGA, LONGC, LAT, FLAT, LONGP, LATP, FLATP, CP0SL, DSAT, DLONG

INTEGER ELL1I, ELL12, ELL13, ELT14, SPOT, ICONTR(16)

EQUIVALENCE(TRYA(1), MSG1(2))

EQUIVALENCE(TRYA(1), MSG1(25))

KEYS = array containing contents of soft keys
PRM = array of values to return for soft keys
RX = actual value returned

DATA ITEM/6NMCCREF/
EXECUTABLE CODE

SPOT = 8
CALL LGBUF(BERBUF,200)
ISTS = LOGU(SSTAT)

read from the header file
REWIND (14)
READ(14,FMT='1622',IOSTAT=ISTAT)POS,FLAT,FLONG,FRADI
CPLOS+POS
CRADI+FRADI
CLAT=FLAT
CLONG=FLONG

IF (CPLOS EQ. POS) THEN
CPLOS+POS
CRADI+FRADI
CLAT=FLAT
CLONG=FLONG
GO TO 368
ENDIF

IF (ISTS NE 8) THEN
WRITE(14,FMT='1635')ISTAT
GO TO 999
ENDIF
GO TO 122
121 POS=999
122 CONTINUE

PAGE=43
IF (CRADI EQ. 1.1) THEN
LINE=27
ELSE
LINE=1
ENDIF

IF (PRIM) THEN
P=3.141592654/180.
R=6378.
IF (PST EQ. 1) THEN
WRITE(UNIT,1001)
WRITE(UNIT,1662)
WRITE(UNIT,1123)
WRITE(UNIT,1124)
CONTINUE
ENDIF
RENIND(16,IOSTAT=ISTAT,ERR=993)
C initialize input buffer with blanks
I=1
CALL FILBS(400,240,MSG,11)
read in record from primary file
191 CONTINUE
READ(18,FMT='1635',IOSTAT=ISTAT,ERR=999,END=193)(MSC(I),I1=1,50)
192 IF (ISTS NE 8) THEN
WRITE(14,FMT='1635')ISTAT
ERRIF
CODE(1,1006,TRY1,LUTLA)
CODE(1,1006,TRY1,LUTLA)
C calculate delta lat, delta long
LATC=LUTLA
LATC=LATC+P
LONGC=CLONG
LONGC=LONGC+P
C if a specific location was specified then
IF (CRADI NE. 1.1) THEN
DIFF=DSTNC(LONGC,CLONG,LATC,LATC)
LATC=R68(LATC-LATT)
LONGC=R68(LATC-LATT)*COS(LATC)
as specific location was specified
ELSE
  DIFFR=0
  DLAT=0
  DLONG=0
ENDIF
WRITE(UNIT,1019)NLIN,LHNE,(HSG1(I1),I1=1,40),DIFFR,DLAT,DLONG
NLIN=NLIN+1
LINE=LINE+1
555
CONTINUE
C
C if end of page, skip to next page
C IF (LINE .GE. PAGE) THEN
  RSPEC=PAGE-LINE
  DO 2 II=1,RSPEC+8
    WRITE(UNIT,1) FORMAT(1)
    CONTINUE
  WRITE(UNIT,1000)
  WRITE(UNIT,1001)
  WRITE(UNIT,1002)
  LINE=5
  NLIN=NLIN+1
ENDIF
225 CONTINUE
C
C if have read all the records for this query then display
the next query location
IF (NLIN .EQ. POS) THEN
  WRITE(UNIT,1)
  WRITE(UNIT,1023)FLAT,FLONC,FRADI
  WRITE(UNIT,1)
  LINE=LINE+5
  CRAD=CRAD+1
  FLAT=FLAT
  FLONG=FLONG
  READ(14,FMT=1022,IOSTAT=ISTAT,END=123)POS,FLAT,FLONG,FRADI
  IF (ISTAT .NE. 0) THEN
    WRITE(UNIT121)ISTAT
    GO TO 999
  ENDIF
  IF (LINE .GE. PAGE) GO TO 555
  ENDIF
GO TO 101
123 POS=POS+999
GO TO 101
ENDIF
183 CONTINUE
IF (SECOND) THEN
  RSPEC=PAGE-LINE
  DO 4 II=1,RSPEC+8
    WRITE(UNIT,1)
    CONTINUE
  NLIN=NLIN+1
ENDIF
C
C if second output is desired then display its headers
IF (SECOND) THEN
  IF (SSQ .EQ. 1) GO TO 184
  SSQ=1
  WRITE(UNIT,1087)
  WRITE(UNIT,1088)
  IF (REALT) .OR. (COS4) THEN
    WRITE(UNIT,1022)
  ELSE
    WRITE(UNIT,1089)
  ENDIF
ENDIF
184 CONTINUE
REWIND(1)
REWIND(14)
READ(14,FMT=1022)POS,FLAT,FLONG,FRADI
CPOS = POS
READ(14,FMT=1022)POS,FLAT,FLONG,FRADI
C
C initialize buffer with blanks
CALL FILPS(400,240,HSG1,II)
SPOT = SPOT + 1
C
C read in record from second file
195 CONTINUE
C read a record from the second display file
IF ((BENT) OR (REAL) OR (CLOSE)) THEN
IF ((CONTR) OR (CLOSE)) THEN
READ(UNIT,201,IOSTAT=ISTAT,ERR=994,END=147)
1 (NL)(1),I(1)=1,52),CONTRY(1),CONTRY(2),IUSER,
1 (TYPE(I11),I(1)=1,4)
ELSE
READ(UNIT,201,IOSTAT=ISTAT,ERR=994,END=147)
1 (NL)(1),I(1)=1,52),CONTRY(1),CONTRY(2),IUSER,
1 ELTI2,ELTI3,ELTI4
ENDIF
ELSE
READ(UNIT,1016,IOSTAT=ISTAT,ERR=994,END=147)(MSG1(I11),I(1)=1,52),
1 ELTI2,ELTI3,ELTI4
ENDIF
994 IF (ISTAT NE. 0) THEN
CONTINUE
WRITE(UNIT,186)ISTAT
GO TO 999
ENDIF
C write a record onto the output logical unit
IF ((BENT) OR (REAL) OR (CLOSE)) THEN
IF ((CONTR) OR (CLOSE)) THEN
WRITE(UNIT,2028),NL,BL,MSG(I11),I(1)=1,52)
1 (CONTRY(1),CONTRY(2),IUSER,(TYPE(I11),I(1)=1,4)
ELSE
WRITE(UNIT,2028),NL,BL,MSG(I11),I(1)=1,52)
1 (CONTRY(1),CONTRY(2),IUSER,ELTI2,ELTI3,ELTI4
ENDIF
ELSE
WRITE(UNIT,1020),NL,MSG(I11),I(1)=1,52),ELTI1,ELTI2,ELTI3,
1 ELTI4
ENDIF
LINE=LINE+1
NL=NL+1
if end of page, skip to next page
IF (LINE .GE. PAGE) THEN
RSAGE-PAGE-LINE
DO 5 I=1,RSAGE
WRITE(UNIT,1)
5 CONTINUE
WRITE(UNIT,1067)
WRITE(UNIT,1068)
WRITE(UNIT,2022)
ELSE
WRITE(UNIT,1069)
ENDIF
LINE=3
NL=NL+1
ENDIF
C see if end of query
C IF (NLNEQ POS) THEN
22 CONTINUE
CPOS = POS
READ(UNIT,201,IOSTAT=ISTAT,ERR=994,END=147)
POS = POS + SPOT + 1
If (POS = SPOT) THEN
GO TO 22
ENDIF
GO TO 145
CONTINUE
POS = POS + 999
GO TO 145
ENDIF
147 IF (THIRD) THEN
RSAGE-PAGE-LINE
DO 6 I=1,RSAGE
WRITE(UNIT,1)
6 CONTINUE
NL=NL+1
6 CONTINUE
IF (THIRD) THEN
WRITE(UNIT,1081)
WRITE(UNIT,1082)
WRITE(UNIT,1063)
CONTINUE
CONTINUE
REINHOMD(2)
C initialize the buffer
111 CALL FILBS(400,240,MSG1,11)
read in record from third file

read a record from the third output file
READ(12,184,ISTAT=ISTAT,ERR=999,END=111)(MSGI(II),II=1,50)
IF (ISTAT .NE. 0) THEN
995 CONTINUE
WRITE(1,110) ISTAT
110 FORMAT(' READ ERROR ON THIRD FILE, ISTAT IS ',I4)
995 CONTINUE
ENDIF
C write a record onto the output logical unit
WRITE(UNIT,182)NLNE,(MSGI(II),II=1,50)
LINE=LINE+1
C if end of pane then skip to the next page
IF (LINE .GE. PAGE) THEN
RSIGE=PAGE-LINE
DO 7 II=1,RSIGE
WRITE(UNIT,1)
7 CONTINUE
LINE=1
ENDIF
GO TO 119
ENDIF
111 CONTINUE
IF (FOURTH) THEN
RSIGE=PAGE-LINE
DO 8 II=1,RSIGE
WRITE(UNIT,1)
8 CONTINUE
LINE=1
ENDIF
GO TO 119
ENDIF
C if the fourth output is desired then display its headers
IF (FOURTH) THEN
IF (FST .EQ. 1) GO TO 113
FST=2
WRITE(UNIT,1815)
WRITE(UNIT,1816)
WRITE(UNIT,1817)
113 CONTINUE
REWIND(13)
C initialize buffer with blanks
114 CALL FILBS(400,246,MSGI(II))
C read in record from fourth file
114 CONTINUE
read a record from the fourth output file
READ(13,1819,ISTAT=ISTAT,ERR=999,END=111)(MSGI(II),II=1,50)
IF (ISTAT .NE. 0) THEN
997 CONTINUE
WRITE(1,115) ISTAT
115 FORMAT(' READ ERROR ON FOURTH FILE, ISTAT IS ',I4)
997 CONTINUE
ENDIF
C write a record onto the output logical unit
WRITE(UNIT,1821)NLNE,(MSGI(II),II=1,50)
LINE=LINE+1
C if end of pane then skip to next page
IF (LINE .GE. PAGE) THEN
RSIGE=PAGE-LINE
DO 9 II=1,RSIGE
WRITE(UNIT,1)
9 CONTINUE
LINE=1
ENDIF
GO TO 114
ENDIF
CONTINUE
C IF (FOURTH) THEN
  IF (.NOT. IF (FOURTH) THEN
    RSPGE=.PACE-L INE
    DO
  CONTINUE
  CONTINUE
  NLINE=3
  ENDIF
C ENDIF
C
C SSS*6OUTPUT IMiAGE DATA
*6*
C IF (IMAGE) THEN
  initialize variables
  PST=0
  SS=0
  TST=0
  FST=0
C
C read from the header file
C REMIND (14)
READ(14 FMT=1822,I0STAT=ISTAT)POS,FLAT,FLONG,FRADI
CPOS=POS
ENDIF
READ(14 FMT=1822,I0STAT=ISTAT,END=721)POS,FLAT,FLONG,FRADI
CPOS=POS
ENDIF
C
C IF (ISTAT NE. 0) THEN
  WRITE(1,720)ISTAT
ENDIF
GO TO 722
720 FORMAT(' ERROR IN READING HEADER FILE, I0S IS ',I4)
GO TO 722
721 POS=999
722 CONTINUE
C
C PAGE=43
C IF (CRADI EQ .0) THEN
  LINE=7
ELSE
  LINE=7
ENDIF
MLINE=1
C
C IF (PRIM) THEN
P=1.4159*254/180.
R=6378.
C IF (PST EQ. 1) GO TO 780
PST=1
WRITE(UNIT,1000)
WRITE(UNIT,1022)
WRITE(UNIT,1002)
WRITE(UNIT,1003)
WRITE(UNIT,1004)CLAT,CLONG,FRADI
WRITE(UNIT,1)
780 CONTINUE
RESEND(21,I0STAT=ISTAT,ERR=1993)
C initialize input buffer with blanks
C II=1
CALL FIELDS(400,248,MSG1,II)
C read in record from primary file
781 CONTINUE
READ(21,FMT=1804,I0STAT=ISTAT,ERR=1993,END=703)
1 (MSG1(II),II=1,50)
C IF (ISTAT NE. 0) THEN
  WRITE(1,782)ISTAT
782 FORMAT(' READ ERROR, ISTAT IS ',I4)
GO TO 999
1993 CONTINUE
WRITE(1,1992)ISTAT
1992 FORMAT(' ERROR IN READING FILE ',I4)
ENDIF
DECIDE(19,1006,TRY1,UL1)
DECIDE(19,1006,TRY1,UL1)
compare to see if this is the correct record

IF (NEVEH LEV) GO TO 580
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 582
GO TO 580

have the correct record

582 CONTINUE

IF (ABS(LUTL-LATA) LT .001) AND
1 (ABS (L UTG-LONGA) LT .001) THEN
1=64
CALL MOVRS(LBUF,55,2,MSG1,II)
MSG1(34)=29A
1=70
CALL FPNOUS(LATB,1777440,MSG1,II)
1=81
CALL FPNOUS(LONGB,1777440,MSG1,II)
1=92
CALL MOVRS(LBUF,57,2,MSG1,II)
MSG1(48)=29B
ELSE
1=64
CALL MOVRS(LBUF,57,2,MSG1,II)
MSG1(34)=29B
1=70
CALL FPNOUS(LATB,1777440,MSG1,II)
1=81
CALL FPNOUS(LONGB,1777440,MSG1,II)
1=92
CALL MOVRS(LBUF,57,2,MSG1,II)
MSG1(48)=29B
ENDIF

580 ENDIF

550 CONTINUE

date

1=2
CALL MOVRS(LBUF,9,6,MSG1,II)
SATPAS & id
1=11+4
CALL MOVRS(LBUF,19,0,MSG1,II)
Mcref
CALL FORM(10,LBUF(4),MSG1,II,6)
event
CALL FORM(10,LBUF(3),MSG1,II,6)
Messnt
1=1+2
CALL FORM(10,LBUF(63),MSG1,II,6)
I-FRQNY
CALL FORM(10,LBUF(25),MSG1,II,4)
eilt lat and alt leng
1=42
CALL FPNOUS(LUTL,1777440,MSG1,II)
1=53
CALL FPNOUS(LUTL,1777440,MSG1,II)
C
IF (OTHMAT) THEN
WRITE(21,10844)MSG1(11),1+1.50
ELSE
WRITE(10,10844)MSG1(11),1+1.50
ENDIF+PSOH+1
END

1000 FORMAT(' PRIMARY DATA')
1001 FORMAT(' SECONDARY')
1002 FORMAT(' DATE SATPAS MCREF EVENT MESSNT ELLONG ELLAT ELLONG DIFF DLAT DLONG')
1004 FORMAT(5942)
1005 FORMAT(' NO ERROR IN MCREF, ISTAT IS +.14')
RETURN
EXECUTABLE CODE

ISYS = LOG(ISYS)

100 CONTINUE

IF (ISYS .EQ. 0) GO TO 550

CALL FINDS(400,240,MSG1,II)

determine if there is an MCCREF number linked in the LUT record

IF (MREF .EQ. 0) GO TO 550

initialize the chain to the MCC detail records

CALL DFGET(IDMCC,ITEM,MREF,1,ISTAT)

IF (ISTAT .EQ. 0) GO TO 550

WRITE(14,1005)ISTAT(1)

read the MCC record

550 CONTINUE

CALL DFGET(IDMCC,5,MREF,ISTAT)

IF (ISTAT .EQ. 0) GO TO 551

WRITE(14,1005)ISTAT(1)

551 CONTINUE
SUBROUTINE PRIMR(OTNDAT)

IMPLICIT NONE

COMMON /LOGG/ LODAT(3), HIDAT(3), STAAT, ENDat, OUTLU(5), RANGE
INTEGER LOOT, HIDAT, OUTLU, RANGE
DOUBLE PRECISION STAAT, ENDat
REAL LOCLAT, LOCLNG, RADIUS, LUTLA, LUTLG

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 IDUTLU(3), IDTST(3), IDSAR(3), IDMCC(3)
INTEGER SDUF(68), LBUF(96), MBUF(7), F7UF(5), RBUF(50)
COMMON /BASE/ IBASE, ILEVEL, IMODE, ISTAT, LIST, ITDTE,
ITTST, ITDAR, ILEVEL, ITSPD, LBUF, MBUF
DOUBLE PRECISION QTIME, SDT
REAL ONEDAY

LOGICAL LUT, MCC, SAN, TEST, ALL
LOGICAL LUT

DATA DECLARATIONS

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

CALLING SEQUENCE:
CALL PRIMR

DATA DECLARATIONS

COMMON /LOCN/ LOCLAT, LOCLNG, RADIUS
COMMON /LOGG/ LODAT(3), HIDAT(3), STAAT, ENDat, OUTLU(5), RANGE
INTEGER LOOT, HIDAT, OUTLU, RANGE
DOUBLE PRECISION STAAT, ENDat
REAL LOCLAT, LOCLNG, RADIUS, LUTLA, LUTLG

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 IDUTLU(3), IDTST(3), IDSAR(3), IDMCC(3)
INTEGER SDUF(68), LBUF(96), MBUF(7), F7UF(5), RBUF(50)
COMMON /BASE/ IBASE, ILEVEL, IMODE, ISTAT, LIST, ITDTE,
ITTST, ITDAR, ILEVEL, ITSPD, LBUF, MBUF
DOUBLE PRECISION QTIME, SDT
REAL ONEDAY

LOGICAL LUT, MCC, SAN, TEST, ALL
LOGICAL LUT

DATA DECLARATIONS

COMMON /LOCN/ LOCLAT, LOCLNG, RADIUS
COMMON /LOGG/ LODAT(3), HIDAT(3), STAAT, ENDat, OUTLU(5), RANGE
INTEGER LOOT, HIDAT, OUTLU, RANGE
DOUBLE PRECISION STAAT, ENDat
REAL LOCLAT, LOCLNG, RADIUS, LUTLA, LUTLG

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 IDUTLU(3), IDTST(3), IDSAR(3), IDMCC(3)
INTEGER SDUF(68), LBUF(96), MBUF(7), F7UF(5), RBUF(50)
COMMON /BASE/ IBASE, ILEVEL, IMODE, ISTAT, LIST, ITDTE,
ITTST, ITDAR, ILEVEL, ITSPD, LBUF, MBUF
DOUBLE PRECISION QTIME, SDT
REAL ONEDAY

LOGICAL LUT, MCC, SAN, TEST, ALL
LOGICAL LUT

DATA DECLARATIONS

COMMON /LOCN/ LOCLAT, LOCLNG, RADIUS
COMMON /LOGG/ LODAT(3), HIDAT(3), STAAT, ENDat, OUTLU(5), RANGE
INTEGER LOOT, HIDAT, OUTLU, RANGE
DOUBLE PRECISION STAAT, ENDat
REAL LOCLAT, LOCLNG, RADIUS, LUTLA, LUTLG

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 IDUTLU(3), IDTST(3), IDSAR(3), IDMCC(3)
INTEGER SDUF(68), LBUF(96), MBUF(7), F7UF(5), RBUF(50)
COMMON /BASE/ IBASE, ILEVEL, IMODE, ISTAT, LIST, ITDTE,
ITTST, ITDAR, ILEVEL, ITSPD, LBUF, MBUF
DOUBLE PRECISION QTIME, SDT
REAL ONEDAY

LOGICAL LUT, MCC, SAN, TEST, ALL
LOGICAL LUT
C FORMATS
1001 FORMAT ("TRACE ERROR - (SELUT) - *",I5)
1002 FORMAT (*,MLT# SAT ID & RCC/SAR MCC FIELD *)
1003 FORMAT (* DEIE PASS NO EVENT INCIDENT REF# TEST LATITUDE - LONGITUDE DISTANCE *)
1004 FORMAT (3X,2A)
1005 FORMAT ("There are ',I4, ' hits.")
1006 FORMAT ("Do you wish a specific countries beacon? ")
1007 FORMAT (A)
1008 FORMAT ("Which country do you wish?")
1009 FORMAT (A)
1010 FORMAT ("Which Canadian beacon(s) do you want?")
1011 FORMAT((11462,13))
999 CONTINUE
C CALLL INCLS(Image,Image,LISTAT)
OPEN(36,FILE='SCRATCH STATUS OLD')
ELSE WRITE(36,1,2)POSN,START,CONTRY(1)
ENDIF
1023 FORMAT(15,1X,15,1X,A2)
CLOSE(30)
CLOSE(1)
CLOSE(12)
CLOSE(4)
CLOSE(2)
CLOSE(26)
CLOSE(23)
CLOSE(24)
CLOSE(UNIT)
CALL EXEC(6)
STOP
END

C*******************************************************************************
C Subroutine to break a single precision integer into its bit string.
C*******************************************************************************
C SUBROUTINE BITSWT(IVAL,BIT)
C INTEGER IVAL,BIT(16),IB,I,J

C BIT(16) = 0
C IB = 16
C IF (IB GE 0) GO TO S
C BIT(16) = 1
C IB = 32768 + IB

C S DO 10 I = 1,15
C J = 16 - I
C BIT(J) = 0
C IF (IB LT 2**21) .LT. 0) GO TO 10
C BIT(J) = 1
C IB = IB - 2**J(3-1)

10 CONTINUE
C RETURN
END
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
Go To 216
ENDIF

END IF

END IF

END IF

CONTINUE

Go To 2

END IF

IF (LBLUF(10) EQ 2H51) OR (LBLUF(16) .EQ. 2H52))H=859.
A = ABS(CTA) * PHI/180.
H = AE * H

ELTANG=SQRT((AE*SIN(A) * H))/((AE+2*(AE-1-COS(A))))
ELTANG=ABS(SIN(A))/SQR((2*SH)*H/(1-COS(A)))*H)
ELTANG = .1 *(ABS(CTA)*HSH/(PHI))

ENDIF

call the output routines according to type of output desired

BOTHAT = .FALSE.
IF (PRIM) CALL PRIMR(OHDAT)
IF (SECOND) CALL SECDT(OHDAT,SCON)
IF (THIRD) CALL TERTIC(OHDAT)
IF (FOURTH) CALL QUAT(OHDAT)
IF (IMAGE) THEN
OHTAT = .TRUE.

CALL CHAIN (IMODE=4)
CALL DQINF (ISASE,IMODE,ISTAT.INFO,14)
IF (ISTAT(I) EQ 0) GO To 124
WRITE (1,100) ISTAT(I)
GO To 999
CONTINUE

124 Get the alternate solution

IMODE = 4
CALL RDGE(TBASE,IMODE,LBLUF,ISTAT,ALTSOL,2)
IF (ISTAT(I) EQ 0) GO To 500
WRITE (1,100) ISTAT(I)
GO To 999
CONTINUE

500 Write out image data

IF (PRIM) CALL PRIMR(OHDAT)
IF (SECOND) CALL SECDT(OHDAT,SCON)
IF (THIRD) CALL TERTIC(OHDAT)
IF (FOURTH) CALL QUAT(OHDAT)

CALL DQINF (IMODE,LBLUF,ISTAT,ALTSOL,2)
IF (ISTAT(I) EQ 0) GO To 125
WRITE (LU.100) ISTAT(I)
GO To 999

125 Restore the chain information

IMODE = 492
CALL DQINF (IMODE,LBLUF,ISTAT,INFO,14)
IF (ISTAT(I) EQ 0) GO To 126
WRITE(LU.100) ISTAT(I)
GO To 999
126 CONTINUE

ENDIF

CONTINUE

DEVENT = LBUF(3)
DLSAT(1)=LBUF(10)
DLSAT(2)=LBUF(11)
DLSAT(3)=LBUF(12)
DLSAT(4)=LBUF(13)
GO To 2
100 CONTINUE

ENDIF

WRITE (LU.1005) (POSNSTART)
IF (LBUF(24) EQ -21203) TYPE(4) = 2H01
IF (LBUF(24) EQ -21207) TYPE(4) = 2H02
IF (LBUF(24) EQ -21208) TYPE(4) = 2H03
IF (LBUF(24) EQ -21209) TYPE(4) = 2H04
IF (LBUF(24) EQ -21210) TYPE(4) = 2H05
IF (LBUF(24) EQ -21211) TYPE(4) = 2H06
IF (LBUF(24) EQ -14243) TYPE(4) = 2H11
IF (LBUF(24) EQ -14247) TYPE(4) = 2H12
IF (LBUF(24) EQ -14252) TYPE(4) = 2H13
IF (LBUF(24) EQ -14246) TYPE(4) = 2H14
IF (LBUF(24) EQ -14251) TYPE(4) = 2H16
GO TO 2
ENDIF

DO I = 1, 11
IF (BEACON(1,1) EQ. 2HCG) THEN
  TYPE(1) = 2HCG
  TYPE(2) = 2HEP
  TYPE(3) = 2HR1
  IF (BEACON(1,4) EQ. 2H01) THEN
    IF (LBUF(24) EQ -21203) THEN
      TYPE(4) = 2H01
      GO TO 2
    ENDIF
  ELSE
    IF (BEACON(1,4) EQ. 2H02) THEN
      IF (LBUF(24) EQ -21207) THEN
        TYPE(4) = 2H02
        GO TO 2
      ENDIF
    ELSE
      IF (BEACON(1,4) EQ. 2H03) THEN
        IF (LBUF(24) EQ -21208) THEN
          TYPE(4) = 2H03
          GO TO 2
        ENDIF
      ELSE
        IF (BEACON(1,4) EQ. 2H04) THEN
          IF (LBUF(24) EQ -21209) THEN
            TYPE(4) = 2H04
            GO TO 2
          ENDIF
        ELSE
          IF (BEACON(1,4) EQ. 2H05) THEN
            IF (LBUF(24) EQ -21210) THEN
              TYPE(4) = 2H05
              GO TO 2
            ENDIF
          ELSE
            IF (BEACON(1,4) EQ. 2H06) THEN
              IF (LBUF(24) EQ -21211) THEN
                TYPE(4) = 2H06
                GO TO 2
              ENDIF
            ELSE
              IF (BEACON(1,4) EQ. 2H07) THEN
                IF (LBUF(24) EQ -14243) THEN
                  TYPE(4) = 2H11
                  GO TO 2
                ENDIF
              ELSE
                IF (BEACON(1,4) EQ. 2H08) THEN
                  IF (LBUF(24) EQ -14247) THEN
                    TYPE(4) = 2H12
                    GO TO 2
                  ENDIF
                ELSE
                  IF (BEACON(1,4) EQ. 2H09) THEN
                    IF (LBUF(24) EQ -14252) THEN
                      TYPE(4) = 2H13
                      GO TO 2
                    ENDIF
                  ELSE
                    IF (BEACON(1,4) EQ. 2H10) THEN
                      IF (LBUF(24) EQ -14246) THEN
                        TYPE(4) = 2H14
                        GO TO 2
                      ENDIF
                    ELSE
                      IF (BEACON(1,4) EQ. 2H11) THEN
                        IF (LBUF(24) EQ -14251) THEN
                          TYPE(4) = 2H16
                          GO TO 2
                        ENDIF
                      ELSE
                        IF (BEACON(1,4) EQ. 2H12) THEN
                          GO TO 2
                        ENDIF
                      ENDIF
                    ENDIF
                  ENDIF
                ENDIF
              ENDIF
            ENDIF
          ENDIF
        ENDIF
      ENDIF
    ENDIF
  ENDIF
ENDIF
Check to see if 486 data is desired

IF ((BENT) OR. (REALT) OR. (COS4)) THEN

Check that have correct country/beacon

CALL BITSC(BUF(21), IBIT)

ICOUNT = IBIT(1)+24*IBIT(2)+48*IBIT(3)+96*IBIT(4)

IF ((IBIT(5)+32*IBIT(6)+48*IBIT(7)+96*IBIT(8))

IF (IBIT(9)+48*IBIT(10)+96*IBIT(11)+IBIT(12))

IF A SPECIFIC COUNTRY IS NOT WANTED DETERMINE THE COUNTRY
OF EACH BEACON

IF (SPECON NE. '1HY') THEN
IF (ICOUNT EQ. 111) THEN
CONTRY(1) = 2HUS
CONTRY(2) = 2HA
ENDIF
IF (ICOUNT EQ. 121) THEN
CONTRY(1) = 2HCA
CONTRY(2) = 2HWA
ENDIF
IF (ICOUNT EQ. 211) THEN
CONTRY(1) = 2HFR
CONTRY(2) = 2HMAN
ENDIF
IF (ICOUNT EQ. 221) THEN
CONTRY(1) = 2HOD
CONTRY(2) = 2HWO
ENDIF
IF (ICOUNT EQ. 241) THEN
CONTRY(1) = 2HUK
CONTRY(2) = 2H4
ENDIF
ENDIF

IF (SPECON EQ. '1HY') THEN
IF ((CONTRY(1) EQ. 2HUS) AND (CONTRY(2) EQ. 2HWA)) THEN
IF (ICOUNT NE. 111) GO TO 200
ELSE
IF (ICOUNT NE. 121) GO TO 200
ELSE
IF (CONTRY(1) EQ. 2HFR) THEN
IF (ICOUNT NE. 211) GO TO 200
ELSE
IF (CONTRY(1) EQ. 2HOD) THEN
IF (ICOUNT NE. 221) GO TO 200
ELSE
IF (CONTRY(1) EQ. 2HUK) THEN
IF (ICOUNT NE. 241) GO TO 200
ENDIF
ENDIF
ELSE
IF (CONTRY(1) EQ. 2HCA) THEN
SEE IF ALL CANADIAN BEACONS ARE DESIRED
IF (BEACON(1) EQ. 2HCA) THEN
IF (ICOUNT (1) EQ. 2H0 ) THEN
IF (LBUF(24) EQ. -21283) OR (LBUF(24) EQ. -21287)
OR (LBUF(24) EQ. -21294) OR (LBUF(24) EQ. -21382)
OR (LBUF(24) EQ. -21311)) THEN
TYPE(1) = 2HCC
TYPE(2) = 2HCP
ENDIF
ELSE
TYPE(1) = 2H0N
TYPE(2) = 2H0U
TYPE(3) = 2HMI
ENDIF
determine if all COSPAS satellites are desired

IF (CX)
  IF ((LBUF(10).NE.2HC1).AND.(LBUF(10).NE.2HC2).AND.
    (LBUF(10).NE.2HC3).AND.(LBUF(10).NE.2HC4)) GO TO 200
ENDIF

determine if a specific satellite is desired

IF (SAT) THEN
  IF ((LDUF(iR).NE.2Hi).AND.(LBUF(10).NE.2HC2)
    AND.(LDF(iR).ME.2HC3).AND.(LBF(iR).NE.2H4)) GO TO 206
ENDIF

determine if a specific SATPAS is desired

IF (SPECiR) THEN
  IF ((LUF(iR).NE.PAST(1)).AND.(LBF(iR).NE.PASS(2))
    AND.(LBUF(iR).NE.PASS(3)).AND.
    (LBUF(iR).NE.PASS(4))) GO TO 260
ENDIF

determine which frequency of ELT is desired for output

CHOSE=.FALSE.

if not all the ELT frequencies are desired then

IF (.NOT.(ALLFR)) THEN
  if CBCl2 and CDC243 are desired then
    IF (CCl2.AND.CPC243) THEN
      IF (LBUF(25).ME.8).AND.(LBUF(25).LE.16)
        GO TO 210
      ENDIF
    GO TO 111
  ENDIF

  if CBCl2 is desired then
    IF (CCl2) THEN
      IF (LBUF(25).ME.8) GO TO 200
    CHOSE=TRUE.
    GO TO 111
  ENDIF

  if CDC243 is desired then
    IF (CDC243) THEN
      IF (LBUF(25).ME.16) GO TO 200
    CHOSE=TRUE.
  ENDIF

if 406 Kbps real time is desired then

IF (REALT) THEN
  IF (LBUF(25).ME.2) GO TO 200
  CHOSE=TRUE.
ENDIF

if 406 'bent pipe' data is desired then

IF (BENT) THEN
  IF (LBUF(25).ME.1) GO TO 200
  CHOSE=TRUE.
ENDIF

if 406 2 Kbps COSPAS stored data is desired then

IF (COS2) THEN
  IF (LBUF(25).ME.4) GO TO 200
  CHOSE=TRUE.
ENDIF

else an ir-lid option was chosen

IF (.NOT.(CHOSE)) THEN
  WRITE(LU,564)
  FORMAT(' AN ILLEGAL FREQUENCY CHOICE WAS MADE. DO AGAIN')
  GO TO 999
ENDIF
ENDIF
111 CONTINUE
Open database

CALL DQPWH(BASE,IBASE,ISTAT)
IF (ISTAT(1) NE 99) WRITE(LU,1025),ISTAT(1)
ENDIF

LU = 1
ONEDAY = -24.0
UNIT = 99
FST = 0
SST = 0
TST = 0

DETERMINE IF A SPECIFIC COUNTRY'S 486 BEACON IS DESIRED

IF ((BFNT) OR (REALT) .OR. (COSA)) THEN
WRITE(LU,1001)
READ(LU,1002),SPECON
IF (SPECON EQ. HY) THEN
WRITE(LU,1002)
READ(LU,1002),SPECON
ENDIF
ENDIF
ENDIF

SET UP THE LOOP COUNTER FOR THE START AND END DATES

DO 199 LOOP = 1,RANGE
II = 1
ONEDAY = ONEDAY + 24.0
SDT = QTIME(LOADAT,II,3,ONEDAY)
II = II + 1
CALL DTINA(SDT,DUM,II)
CALL M035(DUM,DATE1,II)

INitalize chain to LUT detail records

IF (SPECIF) THEN
CALL DQFND(IBASE,IDLUT,5,DATE1,3,ISTAT)
IF (ISTAT(1) EQ. 0) GO TO 100
ELSE
CALL DQFND(IBASE,IDLUT,5,DATE1,3,ISTAT)
IF (ISTAT(1) EQ. 0) GO TO 100
ENDIF

READ THE LUT RECORDS

200 CONTINUE
CALL DQFND(IBASE,IDLUT,5,DATE1,3,ISTAT)
IF (ISTAT(1) EQ. 0) GO TO 100
IF (ISTAT(1) EQ. 156) GO TO 100
WRITE(LU,1001),ISTAT(1)
ENDIF

PROCESS THE RECORD

300 CONTINUE
IF ( (EVENT .EQ. LBUF(1)) .AND. (OLDSAT(1) .EQ. LBUF(10))
    .AND. (OLDSAT(2) .EQ. LBUF(11)) .AND. (OLDSAT(3) .EQ. LBUF(12)) .AND. (OLDSAT(4) .EQ. LBUF(13)))
    GO TO 299

IF NOT ALL THE LOCATIONS DESIRED THEN
IF (.NOT.(ALLUT)) THEN
Determine if this record is in the desired radius
DIF = DSTNC(LOCLNC,LOCLAT,LUTLC,LUTLA)
IF (DIF LT RADIUS) GO TO 299
ENDIF

DETERMINE IF ALL SARSAT SATELLITES ARE DESIRED

IF (5X) THEN
IF ((LBUF(10) NE 2HS1) .AND. ((LBUF(10) NE 2HS2) .AND. (LBUF(10) NE 2HS3) .AND. (LBUF(10) NE 2HS4))) GO TO 299
ENDIF
COMMON /LOG/ LOBAT, HIDAT, STAT, ENDAT, OUTLU, RANGE
COMMON /LOCLAT/ LOCALLAT, LOCLNG
COMMON /RADIUS/MH
COMMON /ILOCCI/ LODAT, 3) NIDAT
COMMON /C/\115 15) PROCRAN
COMMON /ALOCS/ C
COMMON /SEF-40-1R/ V11 (041731.0906)

INTEGER LOYAT, NIDAT, CIJTLU, A.

DOUBLE PRECISION STOAT, ENDAT

COMMON /LOCN/ LOC1AT, \110000
COMMON /IDASE/ S, ILEUL, 3), IMODE

INTEGER ITDTE, 3), ITTST, 3), ITSAR, 3), ITCC, 3), IMAWD

INTEGER IDLUT, 3), IDTST, 3), IDSAR, 3), IDTST

INTEGER SBUE, 69), L3Lff, 99), MRUF, 76), FDUF, 5

COMMON /BASE/ IPfASK, ILEVI, IMODE, ISTAT, LIST, ITDTE

INTEGER IDLUt, IDM~t, IDSAR, IDTST, I3WF, 13W, IST, PRIM, SECOND, THIRD

LOGICAL ALLUT, ALSAT, SAT

INTEGER OEVENTOLDSAT, 1) PLACE, CONTY, 3), BRACON, 11

INTEGER RECGION, CASENO, ICOUNR, IUSEA.

REAL TANG, THETA, EPSAT, I, \110000

INTEGER LOOP, 1, 1, DUM, 7), DATEI, 3)

INTEGER LU, PST, MSCI, UNIT

INTEGER LU\110000

FSET, FIRST time flag

HC1 - output line buffer

UNIT - Fortran LU set to 99

COMMON/SEL1C/\110000

INTEGER ALUT, ASAT, SX, CX, SPESAT, PRIM, SECOND, THIRD, FOURTH

LOGICAL BENT, REAT, COS4, CIRC24, ALFR, CHOSE, EXIT, TAPE

LOGICAL IMAGE, OTHDAT

INTEGER SPESAT, PST, S1T, POST, START, SEVENT, PASS, IMAGE

INTEGER PLACE, CONTRY(3), BEACON(11, 4), SPECM, IBIT(16)

INTEGER REGION, CASENO, SECOND, IUSER, I, TYPE(4)

REAL ELTANG, PH1, NE, HA, 81, CTA

INTEGER DEVENT, OLuS, L19

LOGICAL ALTDIL, LBUF(34)

EQUIVALENCE (LUTL, LBUF(29))

EQUIVALENCE (LUTL, LBUF(33))

EQUIVALENCE (ALTDIL, LBUF(64))

EQUIVALENCE (CTA, LBUF(34))

initialize variables

PH1 = 3.14159265
NE = 9578, 145
H = 1888

OPEN(38, FILE='SCRATCH', STATUS='OLD')
READ (38, \110000

1008 FORMAT(5(I5, RX, SX, CX))

1019 FORMAT(5(I5, RX, SX, CX))

1028 FORMAT(5(I5, RX, SX, CX))

Call OUTP2(PLACE)
IF end of page then skip to next page

IF (LINE > PAGE) THEN
  RSPGE=PAGE-LINE
  DO 99 I=1,RSPGE+8
    WRITE(UNIT,1)
  CONTINUE
  WRITE(UNIT,1015)
  WRITE(UNIT,1016)
  WRITE(UNIT,1017)
LINE=1
END
  END
  ENDIF
GO TO 714

IF (FOURTH) THEN
  RSPGE=PAGE-LINE
  DO 21 I=1,RSPGE+8
    WRITE(UNIT,1)
  CONTINUE
  WRITE(UNIT,115)
  WRITE(UNIT,116)
  WRITE(UNIT,117)
LINE=3
END
ENDIF

1000 FORMAT(35X, PRIMARY DATA*)
1001 FORMAT(35X, PRIMARY SECONDARY*)
1002 FORMAT(35X, PRIMARY LOCATION*)
1003 FORMAT(35X, PRIMARY LOCATION*)
1004 FORMAT(35X, PRIMARY LOCATION*)
1005 FORMAT(35X, PRIMARY LOCATION*)
1006 FORMAT(35X, PRIMARY LOCATION*)
1007 FORMAT(35X, PRIMARY LOCATION*)
1008 FORMAT(35X, PRIMARY LOCATION*)
1009 FORMAT(35X, PRIMARY LOCATION*)
1010 FORMAT(35X, PRIMARY LOCATION*)
1011 FORMAT(35X, PRIMARY LOCATION*)
1012 FORMAT(35X, PRIMARY LOCATION*)
1013 FORMAT(35X, PRIMARY LOCATION*)
1014 FORMAT(35X, PRIMARY LOCATION*)
1015 FORMAT(35X, PRIMARY LOCATION*)
1016 FORMAT(35X, PRIMARY LOCATION*)
1017 FORMAT(35X, PRIMARY LOCATION*)
1018 FORMAT(35X, PRIMARY LOCATION*)
1019 FORMAT(35X, PRIMARY LOCATION*)
1020 FORMAT(35X, PRIMARY LOCATION*)
1021 FORMAT(35X, PRIMARY LOCATION*)
1022 FORMAT(35X, PRIMARY LOCATION*)
1023 FORMAT(35X, PRIMARY LOCATION*)
1024 FORMAT(35X, PRIMARY LOCATION*)
1025 FORMAT(35X, PRIMARY LOCATION*)
1026 FORMAT(35X, PRIMARY LOCATION*)
1027 FORMAT(35X, PRIMARY LOCATION*)
1028 FORMAT(35X, PRIMARY LOCATION*)
1029 FORMAT(35X, PRIMARY LOCATION*)
1030 FORMAT(35X, PRIMARY LOCATION*)
1031 FORMAT(35X, PRIMARY LOCATION*)
1032 FORMAT(35X, PRIMARY LOCATION*)
1033 FORMAT(35X, PRIMARY LOCATION*)
1034 FORMAT(35X, PRIMARY LOCATION*)
1035 FORMAT(35X, PRIMARY LOCATION*)
1036 FORMAT(35X, PRIMARY LOCATION*)
1037 FORMAT(35X, PRIMARY LOCATION*)
1038 FORMAT(35X, PRIMARY LOCATION*)
1039 FORMAT(35X, PRIMARY LOCATION*)
1040 FORMAT(35X, PRIMARY LOCATION*)
1041 FORMAT(35X, PRIMARY LOCATION*)
1042 FORMAT(35X, PRIMARY LOCATION*)
1043 FORMAT(35X, PRIMARY LOCATION*)
1044 FORMAT(35X, PRIMARY LOCATION*)
1045 FORMAT(35X, PRIMARY LOCATION*)
1046 FORMAT(35X, PRIMARY LOCATION*)
1047 FORMAT(35X, PRIMARY LOCATION*)
1048 FORMAT(35X, PRIMARY LOCATION*)
1049 FORMAT(35X, PRIMARY LOCATION*)
1050 FORMAT(35X, PRIMARY LOCATION*)
1051 FORMAT(35X, PRIMARY LOCATION*)
1052 FORMAT(35X, PRIMARY LOCATION*)
1053 FORMAT(35X, PRIMARY LOCATION*)
1054 FORMAT(35X, PRIMARY LOCATION*)
1055 FORMAT(35X, PRIMARY LOCATION*)
1056 FORMAT(35X, PRIMARY LOCATION*)
1057 FORMAT(35X, PRIMARY LOCATION*)
1058 FORMAT(35X, PRIMARY LOCATION*)
1059 FORMAT(35X, PRIMARY LOCATION*)
1060 FORMAT(35X, PRIMARY LOCATION*)
1061 FORMAT(35X, PRIMARY LOCATION*)
1062 FORMAT(35X, PRIMARY LOCATION*)
1063 FORMAT(35X, PRIMARY LOCATION*)
1064 FORMAT(35X, PRIMARY LOCATION*)
1065 FORMAT(35X, PRIMARY LOCATION*)
1066 FORMAT(35X, PRIMARY LOCATION*)
1067 FORMAT(35X, PRIMARY LOCATION*)
1068 FORMAT(35X, PRIMARY LOCATION*)
1069 FORMAT(35X, PRIMARY LOCATION*)
1070 FORMAT(35X, PRIMARY LOCATION*)
1071 FORMAT(35X, PRIMARY LOCATION*)
1072 FORMAT(35X, PRIMARY LOCATION*)
1073 FORMAT(35X, PRIMARY LOCATION*)
1074 FORMAT(35X, PRIMARY LOCATION*)
1075 FORMAT(35X, PRIMARY LOCATION*)
1076 FORMAT(35X, PRIMARY LOCATION*)
1077 FORMAT(35X, PRIMARY LOCATION*)
1078 FORMAT(35X, PRIMARY LOCATION*)
1079 FORMAT(35X, PRIMARY LOCATION*)
1080 FORMAT(35X, PRIMARY LOCATION*)
1081 FORMAT(35X, PRIMARY LOCATION*)
1082 FORMAT(35X, PRIMARY LOCATION*)
1083 FORMAT(35X, PRIMARY LOCATION*)
1084 FORMAT(35X, PRIMARY LOCATION*)
1085 FORMAT(35X, PRIMARY LOCATION*)
1086 FORMAT(35X, PRIMARY LOCATION*)
1087 FORMAT(35X, PRIMARY LOCATION*)
1088 FORMAT(35X, PRIMARY LOCATION*)
1089 FORMAT(35X, PRIMARY LOCATION*)
1090 FORMAT(35X, PRIMARY LOCATION*)
1091 FORMAT(35X, PRIMARY LOCATION*)
1092 FORMAT(35X, PRIMARY LOCATION*)
1093 FORMAT(35X, PRIMARY LOCATION*)
1094 FORMAT(35X, PRIMARY LOCATION*)
1095 FORMAT(35X, PRIMARY LOCATION*)
1096 FORMAT(35X, PRIMARY LOCATION*)
1097 FORMAT(35X, PRIMARY LOCATION*)
1098 FORMAT(35X, PRIMARY LOCATION*)
1099 FORMAT(35X, PRIMARY LOCATION*)
999 CONTINUE
RETURN
END
if the third output is desired then output its headers

IF (THIRD) THEN
  IF (FST EQ 1) GO TO 708
  FST = 1
  WRITE(UNIT,1015)
  WRITE(UNIT,119)

708 CONTINUE

initialize the buffer

I1 = 1
CALL FILBS(400,240,MSG1,11)

read in record from third file

709 CONTINUE

read a record from the third output file

READ(23,1014,IOSTAT=ISTAT,ERR=1995,END=713)(NSG(I),I=1,50)

IF (ISTAT .NE. 0) THEN
  1995 CONTINUE
  WRITE(11,713)ISTAT
  713 FORMAT (READ ERROR ON THIRD FILE, ISTAT IS ',14)
  GO TO 999
ENDIF

write a record onto the output logical unit

WRITE(UNIT,1021)NLINE,(NSG(I),I=1,50)
LINE=LINE+1
NLINE=NLINE+1

if end of page then skip to the next page

IF (LINE .GE. PAGE) THEN
  RSPACE=PAGE-LINE
  DO 777 I11=RSPACE+9
    WRITE(NUNIT,1)
  777 CONTINUE
  LINE=3
  NLINE=I+1
ENDIF

ENDIF

if the fourth output is desired then display its headers

IF (FOURTH) THEN
  IF (FST EQ 1) GO TO 713
  FST = 1
  WRITE(UNIT,1015)
  WRITE(UNIT,119)

713 CONTINUE

initialize buffer with blanks

I1 = 1
CALL FILBS(400,240,MSG1,11)

read in record from fourth file

714 CONTINUE

read a record from the fourth output file

READ(24,1018,IOSTAT=ISTAT,ERR=1997,END=716)(NSG(I),I=1,50)

IF (ISTAT .NE. 0) THEN
  1997 CONTINUE
  WRITE(11,716)ISTAT
  716 FORMAT (READ ERROR ON FOURTH FILE, ISTAT IS ',14)
  GO TO 999
ENDIF

write a record from the output file

WRITE(UNIT,1021)NLINE,(NSG(I),I=1,50)
LINE=LINE+1
NLINE=NLINE+1
initialize buffer with blanks

CALL FILES(400,240,MSG1,I1)

read in record from second file

CALL FIELDS(40D,240,MSG1,I1)

read a record from the second display file

IF (IEND) OR (REAL) OR (COS) THEN
   IF ((CON((SPOT) EQ.2)) THEN
      READ(22,240,MSG1,I1,STAT=1994,END=797)
      ELSE
         READ(22,241,MSG1,I1,STAT=1994,END=797)
   ENDIF
   ELSE
      READ(22,241,MSG1,I1,STAT=1994,END=797)
   ENDIF
ENDIF

write a record onto the output logical unit

IF (IEND) OR (REAL) OR (COS) THEN
   IF (IEND) OR (REAL) OR (COS) THEN
      WRITE(UNIT,2021)I1NSC1(I),IUSER,ELT1,ELT2,ELT3,ELT4
   ELSE
      WRITE(UNIT,2021)I1NSC1(I),IUSER,ELT1,ELT2,ELT3,DELT4
   ENDIF
ELSE
   WRITE(UNIT,1082)I1NSC1(I),IUSER,ELT1,ELT2,ELT3,ELT4
ENDIF

if end of page, skip to next page

IF (LINE.EQ.PAGE) THEN
   WRITE(UNIT,1)
ENDIF

if this is first page

WRITE(UNIT,1)

SEE IF END OF QUERY

IF (LINE.EQ.POS) THEN
   CONTINUE
   IF (PAGE.PAGE) THEN
      WRITE(UNIT,1)
      Go To 795
   ENDIF
ENDIF

CONTINUE

IF (PAGE.PAGE) THEN
   WRITE(UNIT,1)
   Go To 795
ENDIF
calculate delta lat, delta long
LATC=UTLAP
LATT=CLATP
LONGC=CLONCP
LONGE=UTLAP

C if a specific location was specified then
IF (CRAD>0.0) THEN
  DIFFR=OCCUCFNGC,CLAT,UTLGLUTLA)
  DLAT=R(QLATT-LATC)
  DLONG=R(CLONGC-LONGE)8COS(LATC)
ELSE
  DIFFR=0
  DLAT=0
  DLONG=0
ENDIF

1555 CONTINUE

C if end of page, skip to next page
IF (LINE.GE.PAGE) THEN
  RSPGE=PAGELINE
  DO 222 I=I+1,RSPCE+8

222 CONTINUE
WRITE(UNIT,019)NLINE,(HSC111),II=1,40),DIFFR,DLAT,DLONG
NLINE=NLINE+1
LINE=LINE+1
CONTINUE

C if second output is desired then display its headers
IF (SECOND) THEN
  IF (SSI.EQ.1) GO TO 714
  SSI=1
  WRITE(UNIT,17)
  WRITE(UNIT,109)
  IF ((BENT) .NE. REALT) OR (COSA) THEN
    WRITE(UNIT,2022)
  ELSE
    WRITE(UNIT,1019)
  ENDIF
704 CONTINUE
REWIND(22)
REWIND(14)
READ(14,FMT=122)POS,FLAT,FLONG,FRADI
ENDIF
GO TO 704

1123 POS=999
GO TO 704
ENDIF
783 CONTINUE
IF (SECOND) THEN
  IF (LINE.GE.PAGE) THEN
    RSPCE=PAGELINE
    DO 444 I=I+1,RSPCE+8

444 CONTINUE
NLINE=NLINE+1
LINE=LINE+1
ENDIF

GO TO 704
SUBROUTINE SECDY(OTXDAT,SPECON) - SEF-48-08 V09 (840727.1241)

IMPLICIT NONE

C SARSAT EVALUATION FACILITY

DATE VERSION DESCRIPTION AUTHOR
03/10/85  88 ---- SUZANNE Y. SLINN

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

CALLING SEQUENCE:
CALLED FROM: ALOCS
CALL SECDY

DATA DECLARATIONS

COMMON /LOC0L, LODAT(3), HIDAT(3), STDAT, ENDAT, OUTLU(5), RANGE
INTEGER LODAT, HIDAT, OUTLU, RANGE, TRY(10)
DOUBLE PRECISION STDAT, ENDAT
C REAL LOCAT, LOCNG, RADIUS, LU, LUTL, BIAS
C INTEGER LODAT, HIDAT, OUTLU, RANGE
C REAL LOCLAT, LOCLNG, RADIUS
C INTEGER IBASE, ILRE, IMODE, ISTAT, LIST, ITDTE, ITTST, ITSAR, ITNCC, ITSPD
C INTEGER IBASE, IBASE, ILEVE, IMODE, ISTAT, LIST, ITDTE, ITTST, ITSAR, ITNCC,
C     IDTST, I9CC, ALL
C REAL SDT, STDAT, ENDAT
C INTEGER IRB0F(54), IPARN(5), IPRF(33)
C INTEGER IU, IR, RET
C INTEGER DO, ISTY, LOGC
C INTEGER BRF(44), IPARN(5), PBRF(33)
C INTEGER RET
C REAL DAY
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
INTEGER SETS(10,9), PROM(10,9)
INTEGER FLG, NA, IFOR, SCUF(200)

FLG - flag to indicate if E =azed
   NA - dummy parameter for IPOR

COMMON/SELCT/ALLUT, ALSAT, SAT, SX, CX, PRIM, SECOND, THIRD,
   FOURTH, PSI, SST, SST, SST, UNIT, BIF, BENT, REALT, COS24, CRB24
   CRB24, ALFR, POSN, START, EXIT, TAPE, SPEC1F, SEVENT, PASS,
   IMAGE, CONTRY, USER, TYPE, ELTANG
INTEGER CONTAT(3), IDUGER, ITEP(4)
REAL, FLAME
LOGICAL ALLUT, ALSAT, SAT, SX, CX, PRIM, SECOND, THIRD, FOURTH,
LOGICAL BENT REALT, COS24, CRB24, ALFR, EXIT, TAPE, SPEC1F
INTEGER SPESAT(5), PST, SST, SST, MSST(88), POSN, START
INTEGER SEVENT, PASS(4), SECOM
LOGICAL IMAGE, OTHNAT
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 SARASAT satellites wanted
CX - flag to indicate if all COSPAS satellites wanted
SPESAT - array that contains the specific satellites desired
EQUIVALENCE(/LUDA - LUBF(20))
EQUIVALENCE(/LUBF(38))
EQUIVALENCE(/BUBF(38))
EQUIVALENCE(/TRY(1), MSST(49))
EQUIVALENCE(TRY(1), MSST(49))
RETS - array containing contents of soft keys
PRON - array of values to return for soft keys
RE - actual value returned

EXECUTABLE CODE

CALL LGDF(IJBUF, B0001)
IDBT = LOGL(IHST)

initialize variables

L00 CONTINUE

initialize buffer with blanks

I1 = 1
CALL FILBS(48, 168, MSG1, 11)
cross track angle in degrees (CTA)

I1 = 2
CALL FPBLS(LBUF(34), 1777440, MSG1, 11)
number of frequency measurements (WFREQ)

I1 = 10
CALL FPRMS(LBUF(42), MSG1, 11, 6)
standard deviation of residuals in Hz (SDEV)

I1 = 14
CALL FPRMS(LBUF(74), 1777440, MSG1, 11)
trend factor of residuals in Hz (TREND)

I1 = 16
CALL FPRMS(LBUF(76), 1777440, MSG1, 11)
quality factor of CBC data (sun angle) (QUAL)

IF 486 DATA, OUTPUT THE ELT ANGLE INSTEAD
IF ((BENT) OR (REALT) OR (COS24)) THEN
   I1 = 14
   CALL FPRMS(ELTANG, 1777440, MSG1, 11)
ELSE
   I1 = 16
   CALL FPRMS(18, LBUF(90), MSG1, 11, 6)
END IF

 probability of true solution, as % (PROB)
I1 = 1
CALL FORN(10, LUBF(48), MSG1, 11, 6)
number of MLS iterations (MNMLS)

1:=[1+1]
CALL FORM(18,LBUF(43),MSG1,11,6)
time of closest approach in hours from the date of AOS (TCA)

1:=[1+1]
CALL FPNBS(LBUF(36),177744B,MSG1,11)
time of AOS (QTIME)

1:=[1+4]
CALL FPNBS(LBUF(8),177744B,MSG1,11)
date of LOS, in hours from date of AOS (LOSTIM)

1:=[1+4]
CALL FPNBS(LBUF(19),177744B,MSG1,11)
ELT frequency bias, expressed Hz (BIAS)

ENCODE(29,123,T)'BIAS
123 FORMAT(F7.3)

EIIfd is written to the file from LBUF(21),LBUF(22),LBUF(23),LBUF(24)

IF 466 DATA, CHANGE THE ELID CODE OUTPUT
IF ((NENT) OR (REALT) OR (COSA)) THEN
IF COUNTRY IS CANADA, OUTPUT BEACON NAME
IF (COUNTRY(EQ. 2HCA) AND (SPECON EQ. 1)) THEN
IF (OTHDAT) THEN
WRITE(22,1884)(MSG1(1),11=52),COUNTRY(1),COUNTRY(2),
1 USER,TYPE(1),TYPE(2),TYPE(3),TYPE(4)
ELSE
WRITE(11,1884)(MSG1(1),11=52),COUNTRY(1),COUNTRY(2),
1 USER,TYPE(1),TYPE(2),TYPE(3),TYPE(4)
ENDIF
ELSE IF (OTHDAT) THEN
WRITE(22,1885)(MSG1(1),11=52),COUNTRY(1),COUNTRY(2),
1 TYPE(1),LBUF(22),LBUF(23),LBUF(24)
ELSE
WRITE(11,1885)(MSG1(1),11=52),COUNTRY(1),COUNTRY(2),
1 TYPE(1),LBUF(22),LBUF(23),LBUF(24)
ENDIF
ENDIF
ELSE IF (OTHDAT) THEN
WRITE(22,1886)(MSG1(1),11=52),LBUF(21),LBUF(22),LBUF(23),
1 LBUF(24)
ELSE
WRITE(11,1886)(MSG1(1),11=52),LBUF(21),LBUF(22),LBUF(23),
1 LBUF(24)
ENDIF
ENDIF
1884 FORMAT(S2A2,2A2,IX,13,IX,4A2)
1885 FORMAT(S2A2,2A2,IX,13,3(IX,10))
1886 FORMAT(S2A2,4(IX,10))
RETURN
END
SUBROUTINE TERTI(OTHDAT)

IMPLICIT NONE

DATA DECLARATIONS

COMMON /LOGC/ LOCAT(3), HIDAT(3), STDAT, ENDAT, OUTLU(S), RANGE
INTERGER LODAT, HIDAT, OUTLU, RANGE, TRY(10)
DOUBLE PRECISION STDAT, ENDAT
REAL LOCLAT, LOCLNG, RADIUS, LUTLA, LUTLGBIAS

INTEGER IBASE(5), ILEV(3), IMODE, ISTAT(10), LIST
INTERGER ITDTE(3), IITTST(3), ITSAR13), ITMCC(3), ITSPD(3)
INTERGER IMLUT(3), IDTST(3), IDSAR(3), IDNCC(3)
INTERGER ISDUF(61), JUF(98), MIKF(70), FDUF(S1)
COMMON /BASE/ IDASE, ILEV, IMODE, ISTAT, IMLUT
- - ITDTE, IITTST, ITSAR, ITMCC, ITSPD, IMLUT, IDASA
- - IDSAR, IDST, IDNCC, IDSAI, IDTST, IMLUT, IDASE, ILEV, IMODE, ISTAT, IMLUT
DOUBLE PRECISION QTME, STD

SRT - start date

LODAT - oscl start date YYYYDD
STDAT - low date in seconds from 1980
ENDAT - high date in seconds from 1980
OUTLU - output device, LU or filename

INTERGER IPAR(5), FRUF(33)
INTERGER LU, NET
INTERGER SYB, SYS, LOCLU

RFUF - receiving buffer
IPAR - return buffer for OPCAR call
FRUF - return buffer from PARSE

REAL ONE DAY
INTERGER DATCH(3), LOOP, DUM(7), IT
INTERGER UNIT

LOGICAL LUT, MEC, SAR, TEST, ALL

LUT - flag to indicate LUT location detail file selected
MEC - flag to indicate MEC location detail file selected
SAR - flag to indicate SARSTAT incident detail file selected
TEST - flag to indicate FIX.S test master file selected
ALL - all files will be searched

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

CALLING SEQUENCE:
CALLED FROM: ALOCS
CALL TERTI

AUTHOR
SAR

93/11/1
SAR

DATE VERSION DESCRIPTION AUTHOR
83/11/1

---

DESCRIPTION:
This subroutine displays the third of the four output files.
INTEGER KEYS(0,9), PRON(0), RX
INTEGER FLC, AA, IFBRK

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

COMMON/SELECT/ALLUT, ALSAT, SAT, SX, CX, PRIN, PTH, SECOND, THIRD,
    FOURTH, FST, IST, 1ST, 2ND, 3RD, 4TH, REAL, REAL1, COSA, COSB,
    CR24, ALFA, POSN, START, EXIT, TAPE, SPECIF, SEVENT, PASH,
    IMAGE, CONRY, USER, TYPE, ELANG

INTEGER CONTRY(4), USER, TYPE(4)
REAL ELANG

LOGICAL ALLUT, ALSAT, SAT, SX, CX, PRIN, SECOND, THIRD, FOURTH
LOGICAL BENT, REAL1, COSA, COSB, ALFA, POSN, EXIT, TAPE, SPECIF

INTEGER SPESAT(5), FST, IST, 1ST, FST, MSGID(64), POSN, START

INTEGER SEVENT, PASH, RE
LOGICAL IMAGE, OTSHAT
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 COSPAS satellites wanted

SPESAT = array that contains the specific satellites desired

EQUIVALENCE(LUTLA, LBUF(20))
EQUIVALENCE(LUTLG, LBUF(35))
EQUIVALENCE(BIAS, LBUF(78))

KEYS = array containing contents of soft keys
PRON = array of values to return for soft keys
RX = actual value returned

EXECUTABLE CODE

ISYS = LOGLU(SYS)
initialize variables

CONTINUE
11+ CALL FINS(148, 164, MSG1, 11)

drift
11+ CALL FPINS(LBUF(48), 1777948, MSG1, 11)

initial estimate of ETA in degrees (CTAI)
11+ CALL FPINS(LBUF(66), 1777948, MSG1, 11)

initial estimate of ETA in seconds (TCAI)
11+ CALL FPINS(LBUF(68), 1777948, MSG1, 11)

initial estimate of ETA in Hz (BIA8)
ENCODE(21, 123, TRY) BIA8

FORMAT(28, 123, TRY) BIA8

major axis of error ellipse in Hz (NAJAX)
11+ CALL FPINS(LBUF(55), 1777948, MSG1, 11)

minor axis of error ellipse in Hz (NJNAX)
11+ CALL FPINS(LBUF(57), 1777948, MSG1, 11)

average of data residuals in Hz (AMEAN)
11+ CALL FPINS(LBUF(72), 1777948, MSG1, 11)

IF (OTSHAT) THEN
  WRITE(23, 1001)(MSG1(11), I1=1, 59)
ELSE
  WRITE(23, 1001)(MSG1(11), I1=1, 59)
ENDIF

FORMAT(59A2)
RETURN
END
SUBROUTINE QUADY(0THDAT)

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

CALLING SEQUENCE:
CALLED FROM: ALOCS
CALL QUAD

DATA DECLARATIONS

COMMON /LOGG/ LODAT, HIDAT, STDAT, ENDAT, OUTLU(S), RANGE
INTEGER LODAT, HIDAT, OUTLU, RANGE, TRY(10)
DOUBLE PRECISION STDAT, ENDAT
REAL LOCLAT, LOCLNG, RADIUS, LUTLA, LUTLG

COMMON /LOCN/ LOCLAT, LOCLNG, RADIUS

COMMON /LOCM/ LOCLAT, LOCLNG, RADIUS

INTEGER IPASE, ILEVEL, INODE, ISTAT(N), LIST
INTEGER IDTST(3), ISTD(3), IDMCC(3)

INTEGER IDUF(3), LU(98)

DOUBLE PRECISION QTIME, SDT

INTEGER DATCH, LOOP, DUM(7)

INTEGER UNIT

LOGICAL LUT, MCC, SAM, TEST, ALL

SDT - start date
LODAT - select start date YYYY
STDAT - low date in seconds from 1980
ENDAT - high date in seconds from 1980
OUTLU - output device, LU or Filename

REAL ONE DAY

INTEGER IBUF(48), IPARM(5), IPARF(33)
INTEGER LU, KEY
INTEGER SY(8), SYS, LOGLU

IPARM - return buffer for IPARM call
IPARF - return buffer for IPARF call

REAL GROUND

INTEGER DATCH(3), LOOP, DUM(7), II

INTEGER UNIT

LOGICAL LUT, MCC, SAM, TEST, ALL

LUT - flag to indicate LUT location detail file selected
MCC - flag to indicate MCC location detail file selected
SAM - flag to indicate SAMSTAT incident detail file selected
TEST - flag to indicate FIELD test master file selected
ALL - all files will be searched
INTEGER KEYS(0,0), FIGN(1), RX
INTEGER FNLG, AA, IFBRK

FLAG - flag to indicate if E used
AM - dummy parameter for IFBRK

COMMON/SELECT/ALLUT, ALSAT, SAT, CX, SPESAT, PRIN, SECOND, THIRD,
FOURTH, PST, SST, PST, UNIT, DIF, BENI, REALI, CSSA, CC10,
CC24, ALLFI, POSS, START, EXIT, TAPE, SPECIF, SEVENT, PASS,
IMAGE, CONTROL, USER, TYPE, ELTANG

INTEGER CONTAT(1), USER(1), TYPE(1)

REAL ELTANG

LOGICAL ALSAT, SAT, CX, PRIN, SECOND, THIRD, FOURTH

LOGICAL FIRST, REALI, CSSA (CC10), CC24, ALLFI, EXIT, TAPE, SPECIF

LOGICAL SEVENT, PASS(1)

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 SBSAT satellites wanted
SPEsAT - array that contains the specific satellites desired

EQUIVALENCE (LUTLA,LBUF(0))
EQUIVALENCE (LUTLF,LBUF(0))

KEYS - array containing contents of soft keys
PRIM - array of values to return for soft keys
RX - actual value returned

*****************************************************************************

EXECUTABLE CODE

*****************************************************************************

ISYS = LOG(I1S)

initialize variables

100 CONTINUE

CALL FISU(480, 164, MSGL.11)

standard deviation of CTA in degrees (VARCTA)

CALL FPNB(LBUF(78), 177744, MSGL.11)

standard deviation of TCA in seconds (VARCTA)

CALL FPNB(LBUF(80), 177744, MSGL.11)

standard deviation of BIAS in Hz (VARBIA)

CALL FPNB(LBUF(82), 177744, MSGL.11)

standard deviation of DRIFT in Hz/min (VARBIA)

CALL FPNB(LBUF(84), 177744, MSGL.11)

correlation of CTA with TCA (CORCT)

CALL FPNB(LBUF(86), 177744, MSGL.11)

correlation of CTA with BIAS (CORCB)

CALL FPNB(LBUF(88), 177744, MSGL.11)

correlation of TCA with DRIFT (CORCB)

CALL FPNB(LBUF(90), 177744, MSGL.11)

correlation of TCA with BIAS (CORTB)

CALL FPNB(LBUF(92), 177744, MSGL.11)

correlation of TCA with DRIFT (CORTB)

CALL FPNB(LBUF(94), 177744, MSGL.11)
correlation of DRAG with DRIFT (COMB)

r = 1
CALL FNMNS(1,1,777449,MSG1,II)

IF (STUBAT) THEN
  WRITE(29,1884)(MSG1(II),II=1,50)
ELSE
  WRITE(13,1884)(MSG1(II),II=1,50)
ENDIF

1884 FORMAT(SAQZ)
RETURN
END
SUBROUTINE SESET (FTHU, RET)
         IMPLICIT NONE

SATELLITE EVALUATION FACILITY

DATE VERSION DESCRIPTION
03/18/81 00       

DESCRIPTION:
This subroutine is called from LOSAT. It prompts
the terminal for the responses for start date, end date,
and output device.

CALLING SEQUENCE:
CALL SESET(FTHU, RET)

ARGUMENTS:
    FTHU - FTH unit number for output file.
    RET - Return code, 0-> normal,
          1-> error

DATA DECLARATION:
COMMON /LOGS/, LODAT(3), HDAT(3), STGAT, ENDAT, OUTLU(I),
       RANGE

INTEGER FTHU, RET
FTHU = Internal FTH unit
RET = Return code
FTHEDF (LODAT, HDAT, INTLU, RANGE)
DOUBLE PRECISION STGAT, ENDAT, GTIME
DOUBLE PRECISION DUAL
         GTIME = start start date FORMAT
         HDAT = access and date FORMAT
         DUAL = output device, LO or Filename
         NAME = name of file to be considered
         RANG = C or range

C
INTEGER BLU, IPAR(5), IRUF(39)
INTEGER SYS, ISYS, LOGLU

C CDEV - Terminal LU
C PTRLU - Printer LU Set 1
C IPAR - Buffer for RPCM
C IRUF - Input buffer for terminal input
C
INTEGER LUERR

C LUERR - Error return for output file
C
INTEGER DAYS(12), DRUF(3)

C DAYS - Data vector for days in each month
C DRUF - Buffer to store current date, YYMMDD, numeric
C
INTEGER LDAT(3), HDATE(3)

C LDAT - Temporary buffer for low date
C HDATE - Temporary buffer for high date
C
INTEGER RET

C RET - Return code for LOGSET
C
INTEGER DRUF(3)

C DRUF - Result buffer for PARSE
C
INTEGER HO, YE, SLSHE, END

C HO - YYHO
C YE - YYYE
C SLSHE - YYYE
C END - 2XEN
C
INTEGER I, J : IFRR

C I, J - keep control variables
C
DATA HO, YE, SLSHE, END (240, 240, 240, 240)
DATA DAYS (31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31)

C C-initialization
C C-initialization
C
ISYS = LOGLU(SYS)
BENV
BENV

C

C Get current date and time from the system
C (numeric)
C
CALL EXE(l), DRUF, DRUF(3)
DRUF(2) = 0
CALL DATE (DRUF(1), DRUF(2), DRUF(3))
DRUF(1) = DRUF(1) + 1900
DRUF(3) = DRUF(5)
C
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C law date
C Defaults to current TMY1000
C
18 CONTINUE
WRITE(*,99008)
READ(*,99102)(LDATE(1),J=1,3),(HDATE(1),J=1,3)
IF (LDATE(1) LT 0) HDATE(1)=0
IF (LDATE(1) LT 0) GO TO 848
C
IF (LDATE(2) EQ 0) LDATE(2)=0
IF (LDATE(2) LT 1 OR LDATE(2) GT 12) GO TO 858
C
IF (LDATE(3) EQ 0) LDATE(3)=0
DAYS(2)=0
IF (LDATE(3) EQ 0) DAYS(3)=0
IF (LDATE(3) LT 0) GO TO 868
C
C Now set up to convert dates to appropriate format
C
II = 1
DO 42 II = 1,3
42 CALL NUMBS(LDATE(1),2,LODAT,11)
II = 1
STDAT = GMT(LDAT,11,1,8,0)
C
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C law date
C Defaults to current TMY1000
C
58 CONTINUE
IF (HDATE(1) EQ 0) HDATE(1)=0
IF (HDATE(1) LT 0 OR HDATE(1) GT 99) GO TO 888
C
IF (HDATE(2) EQ 0) HDATE(2)=0
IF (HDATE(2) LT 1 OR HDATE(2) GT 12) GO TO 898
C
IF (HDATE(3) EQ 0) HDATE(3)=0
DAYS(2)=0
IF (HDATE(3) EQ 0) DAYS(3)=0
IF (HDATE(3) LT 0) GO TO 908
C
C Now set up to convert dates and times to appropriate format
C
II = 1
DO 42 II = 1,3
42 CALL WDATE(MDATE(1),2,MTAT,24)
II = 1
WDATE = GMT(MDAT,24,1,3,8,0)
C
C Check that high date is beyond low date
C
IF (MTAT LT STDAT) GO TO 978
DAYS = (MTAT-STDAT)/3600 8761+1
HDATE = DPHG
C
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C 188 CONTINUE
C
C Request output device
C
189 CONTINUE
WRITE(*,99011)
READ(*,99029) INUF
CALL PARSE(INUF,0B,0B91)
IF (INUF(2) EQ SLASH OR INUF(2) EQ END) GO TO 981
141
CALL WDATE(INUF,0B,1,91)
IF (EEUF(1) EQ 0) &EEUF(1)=DEV
  DO 118 I=1,3
118 OUT(J)=EEUF(I)
  OUTU1=OUTU1
  OUTU2=OUTU2
  IF (OUTU1 > 770) THEN
    CALL CHUNDD(EEUF(1),OUTU1)
    OUTU1=2M
    OUTU2=2M
    CONTINUE
  END

C
C
C 198 CONTINUE
  continue
  WRITE(DEV,9005) OUTLU,ERR=ERR,OSTAT(1)
  GO TO 999

C
C
C 199 CONTINUE
  continue
  WRITE(DEV,9006) OUTLU,ERR=ERR,OSTAT(2)
  GO TO 999

C
C
C 200 CONTINUE
  continue
  WRITE(DEV,9007) OUTLU,ERR=ERR,OSTAT(3)
  GO TO 999

C
C
C 201 CONTINUE
  continue
  WRITE(DEV,9008) OUTLU,ERR=ERR,OSTAT(4)
  GO TO 999

C
C
C 202 CONTINUE
  continue
  WRITE(DEV,9009) OUTLU,ERR=ERR,OSTAT(5)
  GO TO 999

C
C
C 203 CONTINUE
  continue
  WRITE(DEV,90010)
  GO TO 10
C
C
C EXIT

900 CONTINUE
RETURN

C
C
C FORMATS!
C
C 9066 FORMAT('Enter date for start and end of search: ',
1 ' (##YMMDD ##YMMDD')
9067 FORMAT('Specify retrieval output device LU _,')
9068 FORMAT('# ERROR & File ".,##2," ".,##6," ".,##6," access error ')
C                 In _ )
9069 FORMAT('# ERROR & Answer YES or NO _)
9070 FORMAT('# ERROR & Disaster called incorrectly')
9071 FORMAT('Enter date for start of search _)
9072 FORMAT('# ERROR & Year incorrect _)
9073 FORMAT('# ERROR & Month incorrect _)
9074 FORMAT('# ERROR & Day incorrect _)
9075 FORMAT('# ERROR & High date precedes low date _)
9076 FORMAT(III2,III2)
END
###This file writes the output files onto the magnetic tape

```
5W.1 IN
ST.15.0
ST.12.C0 /E.5
ST.25.0
ST.36.0
ST.45.R
ST.52.0
CA.1
CA.2
CA.3
CA.4
CA.5
```
PROCEDURE :LOCAT

16 SEP. 1983

PROGRAMMER SUZANNE Y. SLINK

VERSION 1.0 (840514 1649)

DESCRIPTION

Lead sequence for LOCAT

---

EC
SZ.29
DP LR
RE ZLOCAT
RE ZDRL
**RE ZAIOLCS
RE EQUP2
**RE ZPRIMARY
**RE ZSECOND
**RE ZTHIRD
**RE ZQUOUTH
RE ZDIFREQUENCY
RE ZJOUTAT
RE ZSPMC
RE ZSEGT
RE ZETST
RE ZNFLL
RE ZSESEL
RE ZSETMC
RE ZCOLWS
SE
SE ZDORLIB
SE ZSPROJ
SE ZISRAS
SEA ZSFLIB
SEA ZRULIB
/E

---

PROCEDURE :ALOCS

14 MAY. 1984

PROGRAMMER SUZANNE Y. SLINK

VERSION 1.0 (840515 1348)

DESCRIPTION

Lead sequence for ALOCS

---
### Title: LOCAL - A DATA RETRIEVAL PROGRAM

### Abstract

The computer program, LOCAL, 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.

LOCAL 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

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 "G" of the DOD 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 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 applicable and the minimum reporting period 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.

8c. 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.

9a. 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) "Distribution is limited to the document from their original documentation center, (2) "Announcement and dissemination of this document is unauthorized 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 be classified "unedited". Each paragraph of the abstract shall contain 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. 73 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.