National Aeronautics and Space Administration
Goddard Earth Science
Data Information and Services Center (GES DISC)

README Document for the Nimbus-3 Satellite Infrared Spectrometer (SIRS) Level 1 Radiance Data

SIRSN3L1

Last Revised 06/26/2019

Goddard Earth Sciences Data and Information Services Center (GES DISC)
https://disc.gsfc.nasa.gov
NASA Goddard Space Flight Center
Code 610.2
Greenbelt, MD 20771 USA

Prepared By:

James E. Johnson	06/26/2019
Name GES DISC GSFC Code 610.2	Date

Reviewed By:

Name	mm/dd/yyyy
Name GSFC Code xxx	Date

Name	mm/dd/yyyy
Name GSFC Code xxx	Date

Goddard Space Flight Center
Greenbelt, Maryland

Revision History

Revision Date	*Changes*	*Author*
06/26/2019	Original	James E. Johnson

Table of Contents

1. Introduction

This document provides basic information on using the Nimbus-3 Satellite Infrared Spectrometer (SIRS) Level-1 Radiance Data product.

1.1 Data Product Description

The Nimbus-3 Satellite Infrared Spectrometer (SIRS) Level-1 Radiance Data product contains radiances that were measured at eight wavelength bands from 11 to 15 microns. Each file contains one orbit worth of data (~14 orbits per day). The SIRS instrument measured at nadir along the orbital track and provided near global spatial coverage (-90 to +90 degrees). The data are available for the days from 14 April 1969 to 19 June 1970. The principal investigators for the SIRS experiment was David Wark from the NOAA National Environmental Satellite, Data and Information Service.

This product was previously available from the NASA National Space Science Data Center (NSSDC) under the name Satellite Infrared Spectrometer Radiance Values with the identifier ESAD-00131 (old id 69-037A-04A).

1.1.1 The Satellite Infrared Spectrometer

The objective of the Nimbus-3 Satellite Infrared Spectrometer (SIRS) experiment was designed to indirectly determine the vertical temperature profiles of the atmosphere by measuring the infrared radiation emitted from the earth and its atmosphere in seven spectral intervals in the carbon dioxide band (13 to 15 micrometers) and one interval in the atmospheric window centered at 11.1 micrometers. The instrument field of view provided data over an area roughly 220 km on a side at a satellite height of 1100 km. Data from the 11.1-micrometer channel yielded surface and/or cloudtop temperatures. Data from the carbon dioxide band could be used to generate temperature-pressure profiles by a mathematical inversion technique.

The SIRS experiment was successful and good data were obtained. On June 21, 1970, the experiment was turned off and all data acquisition effort was transferred to the follow-on SIRS experiment on Nimbus 4.

1.1.2 Nimbus-3 Overview

The Nimbus-3 satellite was successfully launched on April 14, 1969. The spacecraft included the following experiments: 1) a Satellite Infrared Spectrometer (SIRS) for determining the vertical temperature profiles of the atmosphere, (2) an Infrared Interferometer Spectrometer (IRIS) for measuring the emission spectra of the earth-atmosphere system, (3) both High- and Medium-

Resolution Infrared Radiometers (HRIR and MRIR) for yielding information on the distribution and intensity of infrared radiation emitted and reflected by the earth and its atmosphere, (4) a Monitor of Ultraviolet Solar Energy (MUSE) for detecting solar UV radiation, (5) an Image Dissector Camera System (IDCS) for providing daytime cloudcover pictures, (6) a Radioisotope Thermoelectric Generator (RTG), to assess the operational capability of radioisotope power for space applications, and (7) an Interrogation, Recording and Location System (IRLS) experiment designed to locate, interrogate, record, and retransmit meteorological and geophysical data from remote collection stations.

The orbit of the satellite can be characterized by the following:

sun-synchronous near-circular orbit of 1075 km (periapsis) and 1135 km (apoapsis)

inclination of 99.91 degrees

period of an orbit is about 107.4 minutes

orbits cross the equator at 26 degrees of longitude separation

1.2 Algorithm Background

The Nimbus-3 SIRS data were generated from the spacecraft telemetry, attitude and orbital data. The data were originally processed on IBM 360 computers using 24-bit words, and were copied to 6250 tapes for archival. Further information on the SIRS instrument and data processing can be found in the Nimbus-3 Users' Guide Section 6.

1.3 Data Disclaimer

The data should be used with care and one should first read the Nimbus-3 User’s Guide, Section 6 describing the SIRS experiment. Users should also review the Nimbus-3 Data Catalog section on SIRS which includes information on data usage and description of the file format. Users should cite this data product in their research.

2. Data Organization

The Nimbus-3 Satellite Infrared Spectrometer (SIRS) data span the time period from April 14, 1969 to June 19, 1970. Each file typically contains one orbit worth of data (~14 orbits per day).

2.1 File Naming Convention

The data product files are named according to the following convention:

<Platform>-<Instrument>_<Level>_<DateTime>_<Orbit>_<TapeNumber>.<Suffix>

where:
o Platform = name of the platform or satellite (Nimbus3)
o) Instrument = name of the instrument and product (SIRS)
o) Level = process level (L1)
o) Date = Data start date and time in UTC in format <YYYY>m<MMDD>t<hhmmss> where

1. YYYY = 4 digit year (1969 – 1970)
2. MM = 2 digit month (01-12)
3. DD = 2 digit day of month (01-31)
4. hh = 2 digit hour of day (00-23)
5. mm = 2 digit minute (00-59)
6. ss = 2 digit second (00-59)

o Orbit = 5 digit orbit number preceded by the letter ‘o’
o TapeNumber = 3 digit number of tape (preceded by ‘DR’ - primary)
o Suffix = the file format (always TAP, indicating tape binary data)

```
File name example: Nimbus3-SIRS_L1_1969m0522t070347_o00510_DR724.TAP
```

2.2 File Format and Structure

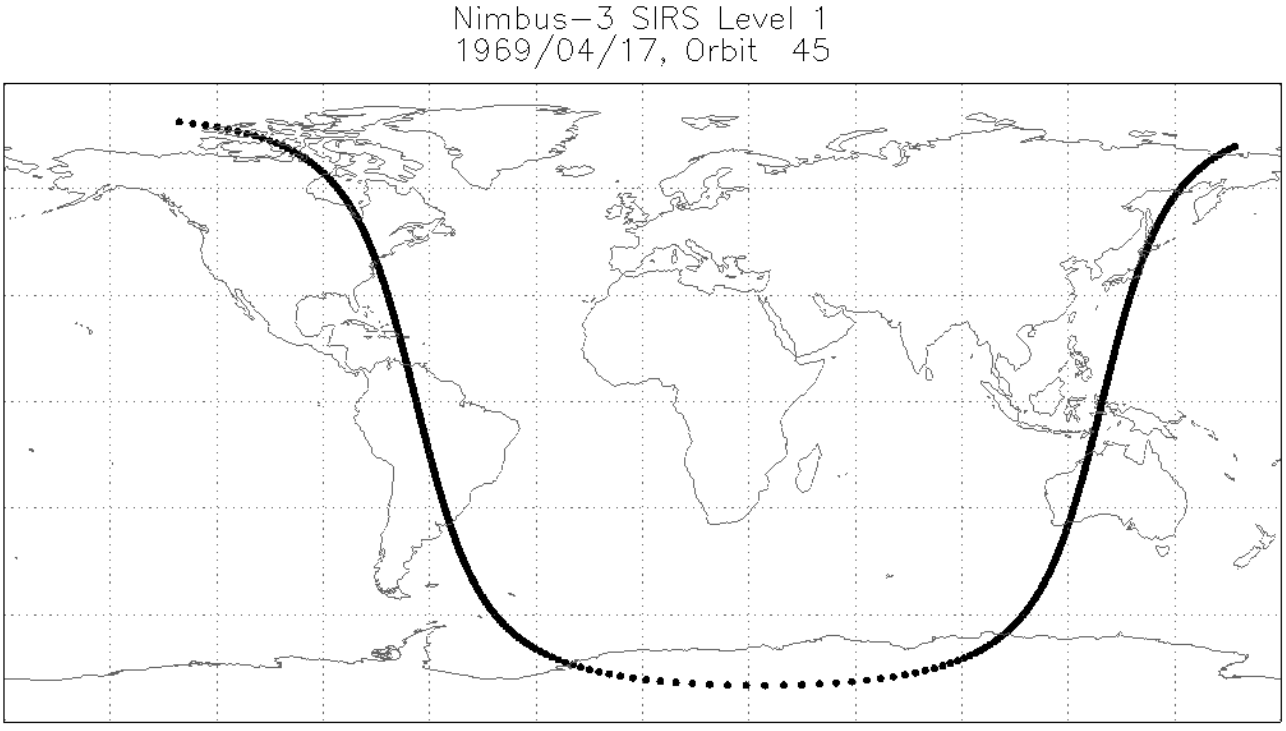
The data are stored as they were originally written in IBM binary (big-endian) record oriented structured files. The files were written to 6250 tapes using a blocked FORTRAN format. The first file on the tape is the tape header file with two records containing text encoded information about the tape. This is followed by up to five orbit files. Each orbit file on the tape contains a header record block and a set of data record blocks with a FORTRAN record size word, the record block, and a FORTRAN record trailing size word. The header record block is 1800 bytes, and contains 450 words of 24-bits. Data record blocks are 4800 bytes, which contain 15 logical data records each consisting of eighty 24-bit words. The original 24-bit words are spread over 32-bit words (four 8-bit words, bits 6 and 7 are not used. There may be up to 375 data records in an orbit file. At the end of the orbit file there is an End-of-File word (the last file on the tape will end with a double End-of-File word). Each data record in a file represents one SIRS measurement with IR radiances for each of the 8 wavelength bands from 11 to 15 microns. Each data file typically contains one orbit worth of data (about 14 orbits per day). For the contents and layout of the data, see section 3.1

During data recovery a total of 5969 data files were retrieved from 6 primary tapes (designated by a DR prefix), there were no backup tapes in the recovery. Caution should be taken as some data records contain corrupted time information (e.g. year is not set to either 69 or 70, and other problems). Five files (orbits 181, 183, 184, 200 and 210) have header records with only 1798 bytes and need 2 zero bytes added at the beginning. One file (orbit 170) has a header record with only 368 bytes which can be padded with zero bytes at the end. One file (orbit 636) has a data record that is only 4790 bytes, pad 10 zero bytes to the end. All of the orbit data files are unique. The Nimbus-3 SIRS files are archived at the GES DISC.

2.3 Key Science Data Fields

The primary science data fields in this data product are the SIRS calibrated radiances in units of erg/(s/cm^2/sr/cm^{-1}) for each of the fourteen IR channels.

Figure 1: Typical Nimbus-3 SIRS Level 1 data file showing data coverage for one orbit.



3. Data Contents

The granularity of this data product is one orbit (with approx. 14 orbits).

3.1 Data Records

The Nimbus-3 User's Guide describes the layout of the file format. There is an update on the time values (packed into one word instead of three) within the "The Nimbus 3 Data Catalog".

The original tape files each included a tape header file. These were then followed by a set of up to 15 orbit data files, As part of the recovery, the GES DISC has extracted and archived the orbit files from the tape. The original data were written on IBM machines using 24-bit words. During tape data recovery these words were spread over four 8-bit bytes (bits 6 and 7 not used). Each data record consists of eighty 24-bit words (see Table 3-1 below), there are typically 15 data records per record block. Each record block starts and ends with a four byte 32-bit integer word giving the block size in bytes.

Table 3-2: Data Record (fifteen 24-bit words)

Word	Field Name		Units	Type	Comments
1 – 30	Orbital Description			30 words	30 BCD characters
31 – 399	SIRS Subsystem Status Profile: (1) Major Frame Number (2-4) Time of Major Frame (Hrs, Min., Sec.) (5) Status of SIRS (4 BCD characters) (6) Status of SOBS (4 BCD characters) (7) Status of SLMP (4 BCD characters) (8) Status of SICM (4 BCD characters) (9) Status of SAT (4 BCD characters) (repeat 41 times, zero filled if not used)			41 x 9 words	
400 401 402 403	Fine Ref. Cone Temperature:	Std. Dev. Min. Max. Mean	°C °C °C °C	1 word 1 word 1 word 1 word	Scaled by 100 Scaled by 100 Scaled by 100 Scaled by 100
404 405 406 407	Course Ref. Cone Temperature:	Std. Dev. Min. Max. Mean	°C °C °C °C	1 word 1 word 1 word 1 word	Scaled by 100 Scaled by 100 Scaled by 100 Scaled by 100
408	Percent. Difference			1 word	Scaled by 100

409 – 411	24 VT:	Min, Max, Mean	V	3 words	Scaled by 100
412 – 414	Motor P.S.:	Min, Max, Mean	V	3 words	Scaled by 100
415 – 417	24 VR:	Min, Max, Mean	V	3 words	Scaled by 100
418 – 420	SCUM Temperature:	Min, Max, Mean	°C	3 words	Scaled by 100
421 – 423	SOBADS Temperature:	Min, Max, Mean	°C	3 words	Scaled by 100
424 – 426	SOD Temperature:	Min, Max, Mean	°C	3 words	Scaled by 100
427 – 429	SIPS Temperature:	Min, Max, Mean	°C	3 words	Scaled by 100
430 – 432	Order Filter Temperature:	Min, Max, Mean	°C	3 words	Scaled by 100
433 – 435	Detector Temperature:	Min, Max, Mean	°C	3 words	Scaled by 100
436 – 438	Calibration Temperature:	Min, Max, Mean	°C	3 words	Scaled by 100
439 – 441	Main Mirror Temperature:	Min, Max, Mean	°C	3 words	Scaled by 100
442 – 444	Motor Temperature:	Min, Max, Mean	°C	3 words	Scaled by 100
445 – 447	Earth Mirror Temperature:	Min, Max, Mean	°C	3 words	Scaled by 100
448 – 450	3 – 24 bit words of ZEROS			3 words	

Table 3-2: Data Record (fifteen 24-bit words)

Word	Field Name	Units	Type	Comments
1	Record Number		1 word	
2	Major Frame Number		1 word	
3	Calibration Code, Time (hour, minute, second)		1 word	4 bytes, each 6 bits
4	Unused		1 word	
5	Unused		1 word	
6	Calibration Cycle Number		1 word	
7	Latitude	degrees	1 word	Scaled by 100
8	Longitude	degrees	1 word	Scaled by 100
9	Altitude	km	1 word	Scaled by 100
10	Attitude	degrees	1 word	Scaled by 100
11 - 26	SIRS IR Data (Bits), Channels 1 – 16	counts	16 words	
27 - 42	SIRS IR Data (Radiance), Channels 1 – 16	mW/cm/sr/	16 words	Scaled by 100

43 – 50	Gain, Channels 1 – 8		8 words	Scaled by 1000
51 – 58	Alpha, Channels 1 – 8		8 words	Scaled by 1000
59	Fine Reference Cone (Bits)	counts	1 word	
60	Fine Reference Cone (Temperature)	°C	1 word	Scaled by 100
61	SCUM Temperature	°C	1 word	Scaled by 100
62	Order Filter Temperature	°C	1 word	Scaled by 100
63	SOBADS Temperature	°C	1 word	Scaled by 100
64	SOD Temperature	°C	1 word	Scaled by 100
65	SIPS Temperature	°C	1 word	Scaled by 100
66	Detector Temperature	°C	1 word	Scaled by 100
67	Calibration Filter Temperature	°C	1 word	Scaled by 100
68	Main Mirror Temperature	°C	1 word	Scaled by 100
69	Motor Temperature	°C	1 word	Scaled by 100
70	24 VT (Voltage)	V	1 word	Scaled by 100
71	Motor P.S. (Voltage)	V	1 word	Scaled by 100
72	24 VR (Voltage)	V	1 word	Scaled by 100
73	Earth Mirror Temperature	°C	1 word	Scaled by 100
74	Course Reference Cone Temperature	°C	1 word	Scaled by 100
75	Status SIRS		1 word	4 BCD characters
76	Status SOBS		1 word	4 BCD characters
77	Status SLMP		1 word	4 BCD characters
78	Status SICM		1 word	4 BCD characters
79	Status SAT		1 word	4 BCD characters
80	Flags (SOLR, LAMP2, SOBSA, SOBSR)		1 word	4 bytes (On/Off)

3.2 Metadata

The metadata are contained in a separate XML formatted file having the same name as the data file with .xml appended to it.

Table 3-2: Metadata attributes associated with the data file.

Name	Description
LongName	Long name of the data product.
ShortName	Short name of the data product.
VersionID	Product or collection version.
GranuleID	Granule identifier, i.e. the name of the file.
Format	File format of the data file.
CheckSumType	Type of checksum used.
CheckSumValue	The value of the calculated checksum.
SizeBytesDataGranule	Size of the file or granule in bytes.
InsertDateTime	Date and time when the granule was inserted into the archive. The format for date is YYYY-MM-DD and time is hh-mm-ss.
ProductionDateTime	Date and time the file was produced in format YYYY-MM-DDThh:mm:ss.ssssssZ
RangeBeginningDate	Begin date when the data was collected in YYYY-MM-DD format.
RangeBeginningTime	Begin time of the date when the data was collected in hh-mm-ss format.
RangeEndingDate	End date when the data was collected in YYYY-MM-DD format.
RangeEndingTime	End time of the date when the data was collected in hh-mm-ss format.
PlatformShortName	Short name or acronym of the platform or satellite
InstrumentShortName	Short name or acronym of the instrument
SensorShortName	Short name or acronym of the sensor
Gpolygon: PointLatitude	Latitudes of the polygon (rectangle) points that represent the satellite coverage. Each point is identified by its latitude and longitude pair.
Gpolygon: PointLongitude	Longitudes of the polygon (rectangle) points that represent the satellite coverage. Each point is identified by its latitude and longitude pair.
Orbit	Orbit number
ElapsedMinTime	Duration in minutes of data collected during an orbit.

4. Reading the Data

The data are written in a binary record-oriented format. Using the record format specification in the section above, users can write software to read the data files. Please note that the data were originally written using a big-endian format, therefore users on little-endian machines will need to swap bytes for the words.

A sample FORTRAN program is included in the Appendix section which will read in the data records. Additionally a FORTRAN function is included to perform byte swapping.

5. Data Services

5.1 GES DISC Search

The GES DISC provides a keyword, spatial, temporal and advanced (event) searches through its unified search and download interface:

https://disc.gsfc.nasa.gov/

5.2 Documentation

The data product landing pages provide information about these data products, as well as links to download the data files and relevant documentation:

https://disc.gsfc.nasa.gov/datacollection/SIRSN3L1_001.html

5.3 Direct Download

These data products are available for users to download directly using HTTPS:

https://acdisc.gesdisc.eosdis.nasa.gov/data/Nimbus3_SIRS_Level1/SIRSN3L1.001/

6. More Information

6.1 Contact Information

Name: GES DISC Help Desk
URL: https://disc.gsfc.nasa.gov/
E-mail: gsfc-help-disc@lists.nasa.gov
Phone: 301-614-5224
Fax: 301-614-5228
Address: Goddard Earth Sciences Data and Information Services Center
Attn: Help Desk
Code 610.2
NASA Goddard Space Flight Center
Greenbelt, MD 20771, USA

6.2 References

"The Nimbus-III User's Guide - Section 6: The Satellite Infrared Spectrometer (SIRS) Experiment", NASA Goddard Space Flight Center, Pages 147-180

"The Nimbus 3 Data Catalog, Volume 1: Section 1.6 The Satellite Infrared Spectrometer (SIRS) Experiment", NASA Goddard Space Flight Center, August 1969, Pages 1-15 to 1-20.

7. Appendices

Acknowledgments

The Nimbus data recovery task at the GES DISC is funded by NASA's Earth Science Data and Information System program.

Acronyms

EOS: Earth Observing System

ESDIS: Earth Science and Data Information System

GES DISC: Goddard Earth Sciences Data and Information Services Center

GSFC: Goddard Space Flight Center

L1: Level-1 Data

NASA: National Aeronautics and Space Administration

QA: Quality Assessment

SIRS: Satellite Infrared Spectrometer

UT: Universal Time

FORTRAN Code

```
C-----------------------------------------------------------------------
C ^NAME: READ_SIRSN3
C   This program will read a Nimbus 3 SIRS Radiance Archival Tape (RAT)
C   Level-1 data file. Each file contains one orbits worth of data.
C
C   The Nimbus 3 SIRS files contain a series of data records. Each of the
C   data records contain the radiances from the 14 SIRS channels, as well as
C   time, geolocation, quality flags and other information. This program will
C   print the contents of each data record.
C
C ^MAJOR VARIABLES:
C     FNAME  - name of input file
C     BUFF   - buffer for data record
C     TEMP   - buffer for holding temporary 4-byte word
C     WORD   - integer 4-byte word
C     IBLKSZ - size of record block in bytes
C     IOS    - I/O status number
C
C ^NOTES:
C     Compile: gfortran -o READ_SIRSN3.EXE READ_SIRSN3.FOR
C
C ^ORGANIZATION: NASA/GSFC, Code 610.2
C
C ^AUTHOR:  James Johnson
C
C ^ADDRESS: james.johnson@nasa.gov
C
C ^CREATED: June 6, 2019
C-----------------------------------------------------------------------

      CHARACTER*256      FNAME          ! Filename
      CHARACTER          BUFF(4800)     ! Buffer for data record block
      INTEGER*4          IBLKSZ         ! Size of records
      INTEGER*4          IWORD          ! 4-byte word
      CHARACTER          TEMP(4)        ! Buffer to hold 4-byte word
      EQUIVALENCE        (TEMP,IWORD)

C Get the name of the input data file to read
      WRITE (0, *), 'Enter the name of the input file:'
      READ (5,'(A)') FNAME
      PRINT '("File = ",A)', FNAME

C Open the specified input file
      OPEN (UNIT=1, FILE=FNAME, STATUS='OLD', ACCESS='DIRECT',
     &      FORM='UNFORMATTED', RECL=1, ERR=99, IOSTAT=IOS)

C Initialize N (record number) and IOFF (byte offset in file)
      N=0
      IOFF=0

C Loop through the file reading all records in file
   10 DO
```

```
C Read the first 4-byte word or record size header
        DO I=1,4
          READ (1, REC=IOFF+I, IOSTAT=IOS, ERR=90) TEMP(I)
        END DO
        IBLKSZ = IWORD
        IOFF=IOFF+(I-1)

C End-of-File (EOF) mark, continue
        IF (IBLKSZ .EQ. 0) GOTO 10

C Next read the block of data
        DO I=1,IBLKSZ
          READ (1, REC=IOFF+I, IOSTAT=IOS) BUFF(I)
          IF (IOS .NE. 0) THEN
            PRINT '("ERROR: BUFF ",I4,X,I4,", IOSTAT: ",I6)', N,I-1,IOS
            IBLKSZ = I-1
            GOTO 20
          END IF
        END DO
        IOFF=IOFF+(I-1)
        N=N+1

C Split data records from record block
        IF (N .EQ. 1) THEN
          CALL PRHREC(IBLKSZ,BUFF)
          PRINT '("-----------------")'
        ELSE
          CALL PRDREC(IBLKSZ,BUFF)
        END IF

C Finally read the last 4-byte word (should match first record size)
   20   DO I=1,4
          READ (1, REC=IOFF+I, IOSTAT=IOS, ERR=90) TEMP(I)
        END DO
        IF (IBLKSZ .NE. IWORD) THEN
          PRINT '("WARNING: IBLKSZ ",I10," != ",I10)', IBLKSZ, IWORD
        ENDIF
        IOFF=IOFF+(I-1)
      END DO

C Close the input file
   90 CLOSE(1)
      GOTO 100

   99 PRINT '("ERROR: OPEN FILE, IOSTAT: ",I6)', IOS

  100 STOP
      END
```

```
C-----------------------------------------------------------------------
C     This Subroutine will print the Header Record
C-----------------------------------------------------------------------
      SUBROUTINE PRHREC(IBLKSZ,BUFF)

      CHARACTER     BUFF(1800)                    ! Buffer for header record
      CHARACTER     DESC*120                      ! String for description
      CHARACTER*4   STRBUF                        ! String buffer
      CHARACTER*4   STAT(5)                       ! Status strings
      CHARACTER     SIRS*4, SOBS*4, SLMP*4        ! Status of SIRS, SOBS, SLMP
      CHARACTER     SICM*4, SAT*4                 ! Status of SICM, SAT
      CHARACTER     TEMP(4)                       ! Temp buffer for word
      INTEGER*4     IWORD                         ! Word value
      REAL*4        STDDEV                        ! Standard Deviation
      REAL*4        MNMXAV(3)                     ! Min, Max, Mean(Avg) Values
      EQUIVALENCE   (TEMP,IWORD)

C     Records should be 1800 bytes, fix incorrect sized headers
      IF (IBLKSZ.EQ.1798) THEN                    ! Pad 2 bytes to beginning
        DO I=IBLKSZ,1,-1
          BUFF(I+2) = BUFF(I)
        END DO
        BUFF(2) = CHAR(0)
        BUFF(1) = CHAR(0)
      ELSE IF (IBLKSZ.EQ.368) THEN                ! Add bytes at end
        DO I=IBLKSZ+1,1800
          BUFF(I) = CHAR(0)
        END DO
      END IF

C     Header description
      DO I = 1,30
        DO J = 1,4
          N = I*4+J-4
          DESC(N:N) = CHAR(IBCD(ICHAR(BUFF(N))))
        END DO
      END DO
      PRINT '("DESC   =",X,A)', DESC

      DO I = 31,450
        N = (I-1)*4+1
        TEMP = BUFF(N:N+3)
        DO M = 1,4
          STRBUF(M:M) = CHAR(IBCD(ICHAR(TEMP(M))))
        END DO
        CALL I24I32(IWORD)
C       STATUS BLOCK
        IF (I.LT.400) THEN
          J = MOD(I-31,9)+1
          K = (I-31)/9+1
          IF      (J.EQ.1) THEN
            MAJFRM = IWORD
          ELSE IF (J.EQ.2) THEN
            IHR = IWORD
          ELSE IF (J.EQ.3) THEN
            IMIN = IWORD
```

```
      ELSE IF (J.EQ.4) THEN
        ISEC = IWORD
      ELSE IF (J.EQ.5) THEN
        STAT(1) = STRBUF
      ELSE IF (J.EQ.6) THEN
        STAT(2) = STRBUF
      ELSE IF (J.EQ.7) THEN
        STAT(3) = STRBUF
      ELSE IF (J.EQ.8) THEN
        STAT(4) = STRBUF
      ELSE IF (J.EQ.9) THEN
        STAT(5) = STRBUF
        IF (K.EQ.1) THEN
          PRINT '("STATUS     MAJFRM     HOUR   MINUTE   SECOND",
     +                X,"SIRS SOBS SLMP SICM  SAT")'
        END IF
        IF (MAJFRM.NE.0) THEN
          PRINT '("(",I2,")   =",4(X,I8),5(X,A4))',
     +        K, MAJFRM, IHR, IMIN, ISEC, STAT
        END IF
      END IF
    ELSE
      IF      (I.EQ.400) THEN
        STDDEV = IWORD/100.
      ELSE IF (I.EQ.401) THEN
        MNMXAV(1) = IWORD/100.
      ELSE IF (I.EQ.402) THEN
        MNMXAV(2) = IWORD/100.
      ELSE IF (I.EQ.403) THEN
        MNMXAV(3) = IWORD/100.
        PRINT '("TFRCON =",4(X,F8.2),X,"(STDDEV,MIN,MAX,MEAN)")',
     +      STDDEV,MNMXAV
      ELSE IF (I.EQ.404) THEN
        STDDEV = IWORD/100.
      ELSE IF (I.EQ.405) THEN
        MNMXAV(1) = IWORD/100.
      ELSE IF (I.EQ.406) THEN
        MNMXAV(2) = IWORD/100.
      ELSE IF (I.EQ.407) THEN
        MNMXAV(3) = IWORD/100.
        PRINT '("TCRCON =",4(X,F8.2),X,"(STDDEV,MIN,MAX,MEAN)")',
     +      STDDEV,MNMXAV
      ELSE IF (I.EQ.408) THEN
        PRINT '("PCTDIF =",X,F8.2)', IWORD/100.
      ELSE IF (I.EQ.409) THEN
        MNMXAV(1) = IWORD/100.
      ELSE IF (I.EQ.410) THEN
        MNMXAV(2) = IWORD/100.
      ELSE IF (I.EQ.411) THEN
        MNMXAV(3) = IWORD/100.
        PRINT '("VT24   =",3(X,F8.2),X,"(MIN,MAX,MEAN)")', MNMXAV
      ELSE IF (I.EQ.412) THEN
        MNMXAV(1) = IWORD/100.
      ELSE IF (I.EQ.413) THEN
        MNMXAV(2) = IWORD/100.
```

```
ELSE IF (I.EQ.414) THEN
  MNMXAV(3) = IWORD/100.
  PRINT '("MOTPS  =",3(X,F8.2),X,"(MIN,MAX,MEAN)")', MNMXAV
ELSE IF (I.EQ.415) THEN
  MNMXAV(1) = IWORD/100.
ELSE IF (I.EQ.416) THEN
  MNMXAV(2) = IWORD/100.
ELSE IF (I.EQ.417) THEN
  MNMXAV(3) = IWORD/100.
  PRINT '("VR24   =",3(X,F8.2),X,"(MIN,MAX,MEAN)")', MNMXAV
ELSE IF (I.EQ.418) THEN
  MNMXAV(1) = IWORD/100.
ELSE IF (I.EQ.419) THEN
  MNMXAV(2) = IWORD/100.
ELSE IF (I.EQ.420) THEN
  MNMXAV(3) = IWORD/100.
  PRINT '("TSCUM  =",3(X,F8.2)," (MIN,MAX,MEAN)")', MNMXAV
ELSE IF (I.EQ.421) THEN
  MNMXAV(1) = IWORD/100.
ELSE IF (I.EQ.422) THEN
  MNMXAV(2) = IWORD/100.
ELSE IF (I.EQ.423) THEN
  MNMXAV(3) = IWORD/100.
  PRINT '("TSOBAD =",3(X,F8.2)," (MIN,MAX,MEAN)")', MNMXAV
ELSE IF (I.EQ.424) THEN
  MNMXAV(1) = IWORD/100.
ELSE IF (I.EQ.425) THEN
  MNMXAV(2) = IWORD/100.
ELSE IF (I.EQ.426) THEN
  MNMXAV(3) = IWORD/100.
  PRINT '("TSOD   =",3(X,F8.2)," (MIN,MAX,MEAN)")', MNMXAV
ELSE IF (I.EQ.427) THEN
  MNMXAV(1) = IWORD/100.
ELSE IF (I.EQ.428) THEN
  MNMXAV(2) = IWORD/100.
ELSE IF (I.EQ.429) THEN
  MNMXAV(3) = IWORD/100.
  PRINT '("TSIPS  =",3(X,F8.2)," (MIN,MAX,MEAN)")', MNMXAV
ELSE IF (I.EQ.430) THEN
  MNMXAV(1) = IWORD/100.
ELSE IF (I.EQ.431) THEN
  MNMXAV(2) = IWORD/100.
ELSE IF (I.EQ.432) THEN
  MNMXAV(3) = IWORD/100.
  PRINT '("TORDFL =",3(X,F8.2)," (MIN,MAX,MEAN)")', MNMXAV
ELSE IF (I.EQ.433) THEN
  MNMXAV(1) = IWORD/100.
ELSE IF (I.EQ.434) THEN
  MNMXAV(2) = IWORD/100.
ELSE IF (I.EQ.435) THEN
  MNMXAV(3) = IWORD/100.
  PRINT '("TDETEC =",3(X,F8.2)," (MIN,MAX,MEAN)")', MNMXAV
ELSE IF (I.EQ.436) THEN
  MNMXAV(1) = IWORD/100.
ELSE IF (I.EQ.437) THEN
  MNMXAV(2) = IWORD/100.
```

```
            ELSE IF (I.EQ.438) THEN
              MNMXAV(3) = IWORD/100.
              PRINT '("TCALFL =",3(X,F8.2)," (MIN,MAX,MEAN)")', MNMXAV
            ELSE IF (I.EQ.439) THEN
              MNMXAV(1) = IWORD/100.
            ELSE IF (I.EQ.440) THEN
              MNMXAV(2) = IWORD/100.
            ELSE IF (I.EQ.441) THEN
              MNMXAV(3) = IWORD/100.
              PRINT '("TMMIRR =",3(X,F8.2)," (MIN,MAX,MEAN)")', MNMXAV
            ELSE IF (I.EQ.442) THEN
              MNMXAV(1) = IWORD/100.
            ELSE IF (I.EQ.443) THEN
              MNMXAV(2) = IWORD/100.
            ELSE IF (I.EQ.444) THEN
              MNMXAV(3) = IWORD/100.
              PRINT '("TMOTOR =",3(X,F8.2)," (MIN,MAX,MEAN)")', MNMXAV
            ELSE IF (I.EQ.445) THEN
              MNMXAV(1) = IWORD/100.
            ELSE IF (I.EQ.446) THEN
              MNMXAV(2) = IWORD/100.
            ELSE IF (I.EQ.447) THEN
              MNMXAV(3) = IWORD/100.
              PRINT '("TEMIRR =",3(X,F8.2)," (MIN,MAX,MEAN)")', MNMXAV
            ELSE IF (I.EQ.448) THEN
              MNMXAV(1) = IWORD/100.
            ELSE IF (I.EQ.449) THEN
              MNMXAV(2) = IWORD/100.
C           ELSE IF (I.EQ.450) THEN
C             MNMXAV(3) = IWORD/100.
C             PRINT '("ZEROES =",3(X,F8.2))', MNMXAV
            END IF
          END IF
        END DO

        RETURN
        END
```

```
C-----------------------------------------------------------------------
C     This Subroutine will print the Data Records
C-----------------------------------------------------------------------
      SUBROUTINE PRDREC(IBLKSZ, BUFF)

      CHARACTER     BUFF(15*4)                      ! Buffer for record block
      CHARACTER     TEMP(4)                         ! Temp buffer for word
      CHARACTER*4   STRBUF                          ! String buffer for word
      CHARACTER*4   STAT(5)                         ! Status strings
      INTEGER*4     IWORD                           ! Word value
      INTEGER*1     CALCOD, THOUR, TMIN, TSEC       ! Code, time (hour, min, sec)
      INTEGER*4     IRBITS(16)                      ! IR Radiance bits
      REAL*4        IRRADS(16)                      ! IR Radiance values
      REAL*4        GAIN(8), ALPHA(8)               ! Gain and alpha
      INTEGER*1     SOLR, LAMP2, SOBSA, SOBSB       ! Flags SOLR,LAMP2,SOBSA/B
      EQUIVALENCE   (TEMP,IWORD)

      NRECS = IBLKSZ/4/80

      DO I = 1,NRECS
        DO J = 1, 80
          N = ((I-1)*80+J)*4-3
          TEMP = BUFF(N:N+3)
          DO M = 1,4
            STRBUF(M:M) = CHAR(IBCD(ICHAR(TEMP(M))))
          END DO
          CALL I24I32(IWORD)
          IF      (J.EQ.1) THEN
            IF (IWORD.EQ.0) EXIT
            PRINT '("RECNUM =",X,I8),")', IWORD
          ELSE IF (J.EQ.2) THEN
            PRINT '("MAJFRM =",X,I8),")', IWORD
          ELSE IF (J.EQ.3) THEN
            CALCOD  = IAND(ISHFT(IWORD,-3*6), '3F'Z)
            PRINT '("CALCOD =",X,I8)', CALCOD
            THOUR  = IAND(ISHFT(IWORD,-2*6), '3F'Z)
            TMIN   = IAND(ISHFT(IWORD,-1*6), '3F'Z)
            TSEC   = IAND(ISHFT(IWORD,-0*6), '3F'Z)
            PRINT '("TIME   =",3(X,I8)," (HOUR,MIN,SEC)")',
     +      THOUR, TMIN, TSEC
C         ELSE IF (J.EQ.4) THEN
C           IZERO1 = IWORD
C         ELSE IF (J.EQ.5) THEN
C           IZERO2 = IWORD
          ELSE IF (J.EQ.6) THEN
            ICALNO = IWORD
            PRINT '("CALNUM =",X,I8),")', IWORD
          ELSE IF (J.EQ.7) THEN
            PRINT '("LAT    =",X,F8.2),")', IWORD/100.
          ELSE IF (J.EQ.8) THEN
            PRINT '("LON    =",X,F8.2),")', IWORD/100.
          ELSE IF (J.EQ.9) THEN
            PRINT '("ALT    =",X,F8.2),")', IWORD/100.
          ELSE IF (J.EQ.10) THEN
            PRINT '("ATT    =",X,F8.2),")', IWORD/100.
```

```
ELSE IF (J.GE.11.AND.J.LT.27) THEN
  IRBITS(J-11+1) = IWORD
  IF (J.EQ.26) THEN
    PRINT '("IRBITS =",8(X,I8),/,8X,8(X,I8))', IRBITS
  END IF
ELSE IF (J.GE.27.AND.J.LT.43) THEN
  IRRADS(J-27+1) = IWORD/100.
  IF (J.EQ.42) THEN
    PRINT '("IRRADS =",8(X,F8.2),/,8X,8(X,F8.2))', IRRADS
  END IF
ELSE IF (J.GE.43.AND.J.LT.51) THEN
  GAIN(J-43+1) = IWORD/1000.
  IF (J.EQ.50) THEN
    PRINT '("GAIN   =",8(X,F8.3)),")', GAIN
  END IF
ELSE IF (J.GE.51.AND.J.LT.59) THEN
  ALPHA(J-51+1) = IWORD/1000.
  IF (J.EQ.58) THEN
    PRINT '("ALPHA  =",8(X,F8.3)),")', ALPHA
  END IF
ELSE IF (J.EQ.59) THEN
  PRINT '("BFRCON =",X,I8),")', IWORD
ELSE IF (J.EQ.60) THEN
  PRINT '("TFRCON =",X,F8.2),")', IWORD/100.
ELSE IF (J.EQ.61) THEN
  PRINT '("TSCUM  =",X,F8.2),")', IWORD/100.
ELSE IF (J.EQ.62) THEN
  PRINT '("TORDFL =",X,F8.2),")', IWORD/100.
ELSE IF (J.EQ.63) THEN
  PRINT '("TSOBAD =",X,F8.2),")', IWORD/100.
ELSE IF (J.EQ.64) THEN
  PRINT '("TSOD   =",X,F8.2),")', IWORD/100.
ELSE IF (J.EQ.65) THEN
  PRINT '("TSIPS  =",X,F8.2),")', IWORD/100.
ELSE IF (J.EQ.66) THEN
  PRINT '("TDETEC =",X,F8.2),")', IWORD/100.
ELSE IF (J.EQ.67) THEN
  PRINT '("TCALFL =",X,F8.2),")', IWORD/100.
ELSE IF (J.EQ.68) THEN
  PRINT '("TMMIRR =",X,F8.2),")', IWORD/100.
ELSE IF (J.EQ.69) THEN
  PRINT '("TMOTOR =",X,F8.2),")', IWORD/100.
ELSE IF (J.EQ.70) THEN
  PRINT '("N24VT  =",X,F8.2),")', IWORD/100.
ELSE IF (J.EQ.71) THEN
  PRINT '("MOTPS  =",X,F8.2),")', IWORD/100.
ELSE IF (J.EQ.72) THEN
  PRINT '("N24VR  =",X,F8.2),")', IWORD/100.
ELSE IF (J.EQ.73) THEN
  PRINT '("TEMIRR =",X,F8.2),")', IWORD/100.
ELSE IF (J.EQ.74) THEN
  PRINT '("TCRCON =",X,F8.2),")', IWORD/100.
ELSE IF (J.EQ.75) THEN
  STAT(1) = STRBUF
ELSE IF (J.EQ.76) THEN
  STAT(2) = STRBUF
```

```
      ELSE IF (J.EQ.77) THEN
        STAT(3) = STRBUF
      ELSE IF (J.EQ.78) THEN
        STAT(4) = STRBUF
      ELSE IF (J.EQ.79) THEN
        STAT(5) = STRBUF
        PRINT '("STATUS =",5(X,A8),X,"(SIRS,SOBS,SLMP,SICM,SAT)")',
     +      STAT
      ELSE IF (J.EQ.80) THEN
        SOLR  = IAND(ISHFT(IWORD,-3*6), '3F'Z)
        LAMP2 = IAND(ISHFT(IWORD,-2*6), '3F'Z)
        SOBSA = IAND(ISHFT(IWORD,-1*6), '3F'Z)
        SOBSB = IAND(ISHFT(IWORD,-0*6), '3F'Z)
        PRINT '("FLAGS  =",4(X,I8),10X,"(SOLR,LAMP2,SOBSA,SOBSB)")',
     +      SOLR, LAMP2, SOBSA, SOBSB
        PRINT '("-----------------")'
      END IF
    END DO
  END DO

  RETURN
  END
```

```
C--------------------------------------------------------------------------
C     This Subroutine will convert 24-bit word to 32-bits
C--------------------------------------------------------------------------
      SUBROUTINE I24I32(IWORD)

      INTEGER*4     I4TEMP                      ! 15 word data record
      CHARACTER     TEMP(4)                     ! Temp buffer for data record
      EQUIVALENCE   (TEMP,I4TEMP)

      I4TEMP = IWORD
      IW = 0
      DO K = 1,4
        IW = ISHFT(IW, 6)                       ! Shift left by 6 bits
        IW = IOR(IW,IAND(ICHAR(TEMP(K)),Z'3F')) ! Remove 2 most signif bits
      END DO
      IF (ISHFT(IW,-23).EQ.1) THEN
        IW = IW - 2**24                         ! Negative value
      END IF

      IWORD = IW

      RETURN
      END

C--------------------------------------------------------------------------
C     This Function will return BCD to ASCII character index
C--------------------------------------------------------------------------
      FUNCTION IBCD(I)

      CHARACTER BCDTBL(64)
C                  _0  _1  _2  _3  _4  _5  _6  _7
      DATA BCDTBL /':','A','B','C','D','E','F','G', ! 0_
     +             'H','I','J','K','L','M','N','O', ! 1_
     +             'P','Q','R','S','T','U','V','W', ! 2_
     +             'X','Y','Z','0','1','2','3','4', ! 3_
     +             '5','6','7','8','9','+','-','*', ! 4_
     +             '/','(',')','$','=',' ',',','.', ! 5_
     +             '#','[',']','%','"','_','!','&', ! 6_
     +             "'",'?','<','>','@','\','^',';'/ ! 7_

      IBCD = ICHAR(BCDTBL(I+1))

      RETURN
      END
```