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

README Document for the
Nimbus-7
Temperature Humidity Infrared Radiometer (THIR)
Level 1 Calibrated Located Radiance Data at
6.7 and 11.5 microns

THIRN7L1CLDT

Last Revised 05/20/2015

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

Prepared By:

James E. Johnson	05/20/2015
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*
05/20/2015	Original	James E. Johnson

Table of Contents

1. Introduction

This document provides basic information on using the Nimbus-7 Temperature Humidity Infrared Radiometer (THIR) Level-1 Calibrated Located Radiation Data at 6.7 and 11.5 microns product.

1.1 Data Product Description

The Nimbus-7 Temperature Humidity Infrared Radiometer (THIR) Level-1 Calibrated Located Radiation Data product contains the calibrated and geolocated radiances of thermal emissions from the earth and atmosphere in the 6.5 – 7.0 (6.7) micron and 10.5 – 12.5 (11.5) micron channels. Unlike the THIR data from previous Nimbus satellite missions, this product contains data from both channels in a single data file. Each file typically contains one full orbit (~104 minutes) worth of data. Spatial coverage is global. The data are available from October 30, 1978 (day of year 303) through May 13, 1985 (day of year 133).

This product was previously available from the NASA National Space Science Data Center (NSSDC) under the name THIR Calibrated Located Radiation Data Tape (CLDT) with the identifier ESAD-00170 (old id 78-098A-10C).

1.1.1 The Temperature Humidity Infrared Radiometer

The THIR instrument is a two channel high resolution scanning radiometer designed to perform two major functions:

- 6.5 – 7.0 (6.7) micron channel gives information on the moisture content of the upper troposphere and stratosphere and the location of jet streams and frontal systems. The water vapor channel has a resolution of the sub-point is 20 km and operates mostly at night.
- 10.5 – 12.5 (11.5) micron channel provides both day and night cloud top or surface temperatures. The ground resolution at the sub-point is 6.7 km and operates day and night.

The optical system of the Nimbus THIR instrument consists of a scan mirror, a telescope, and a dichroic beam splitter. The scan mirror is inclined to 45 degrees to the axis of rotation (scans perpendicular to flight path) and the scan rate operation is 48 revolutions per minute. The field of view scans across the earth from east to west in daytime and west to east at night when traveling northward and southward respectively. A dichroic beam splitter divides the energy into two channels. A 20 milliradian channel detects energy in the 6.7 micron band while a 7.0

milliradian channel detects energy in the 11.5 micron band. In both cases a germanium immersed thermistor bolometer is used. The swath width is about 2600 km

The Nimbus-7 THIR instrument was basically the same as previous THIR flown on the Nimbus-4, 5 and 6 satellites, except that the onboard system was digitized. The experiment was successful returning data until 1985 when it was turned off to conserve spacecraft power.

The principal investigator for the THIR experiment was Dr. Larry L. Stowe from NOAA NESDIS.

1.1.2 Nimbus-7 Overview

The Nimbus-7 satellite was successfully launched on October 24, 1978 and was the final in the Nimbus series. The spacecraft included nine experiments: (1) the Limb Infrared Monitor of the Stratosphere (LIMS) for making vertical profiles of temperature and concentrations of O_3, H_2O, NO_2, and HNO_3, (2) a Stratospheric and Mesospheric Sounder (SAMS) providing vertical concentrations of H_2O, CH_4, CO and NO and measure the temperature in the upper atmosphere, (3) the Coastal-Zone Color Scanner (CZCS) for mapping ocean chlorophyll concentrations, (4) the Stratospheric Aerosol Measurement II (SAM II) to map the concentration and optical properties of aerosols, (5) the Earth Radiation Budget (ERB) for measuring the incoming and outgoing reflected and emitted radiation of the Earth, (6) a Scanning Multichannel Microwave Radiometer (SMMR) to obtain and use ocean momentum and energy-transfer parameters on a nearly all-weather operational basis., (7) a Solar Backscatter UV (SBUV) spectrometer to determine the vertical distribution of ozone, (8) the Total Ozone Mapping Spectrometer (TOMS) for mapping the total column amount of ozone, and (9) the Temperature Humidity Infrared Radiometer (THIR) for measuring daytime and nighttime surface and cloudtop temperatures, as well as the water vapor content of the upper atmosphere.

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

- circular orbit at ~950 km
- inclination of 99 degrees
- period of an orbit is about 104 minutes
- orbits cross the equator at 26 degrees of longitude separation
- sun-synchronous

1.2 Algorithm Background

The Nimbus-7 THIR data were generated from the spacecraft telemetry, attitude and orbital data. The data were originally processed on IBM 360 computers using a 32-bit architecture, and copied to 1600 bpi 9-track tapes for archival. Further information on the THIR instrument and data processing can be found in the Nimbus-7 Users' Guide Section 9 and the Nimbus 7 Temperature-Humidity Infrared Radiometer (THIR) Data User's Guide.

1.3 Data Disclaimer

The data should be used with care and one should first read the Nimbus-7 User's Guide, section 9 describing the THIR experiment. Users should cite this data product in their research.

2. Data Organization

The Nimbus-7 Temperature Humidity Infrared Radiometer Level-1 Calibrated Located Radiation Data (CLDT) spans the time period from October 30, 1978 to May 9, 1985. Each file typically contains about one full orbit (104 minutes) worth of data.

2.1 File Naming Convention

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

<Platform>_<Instrument/Product>_<Date>_<OrbitNumber>-<TapeNumber>.<Suffix>

where:
o Platform = name of the platform or satellite (always Nimbus7)
o) Instrument/Product = name of the instrument and product (always THIRCLDT)
o Date = Data start date and time in UTC in format <YYYY>m<MMDD>t<hhmmss> where

1. YYYY = 4 digit year (1978 – 1985)
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 of hour (00-59)
6. ss = 2 digit seconds of hour (00-59)

o OrbitNumber = number of orbit when the data were collected (preceded by the letter 'o')
o TapeNumber = number of tape (preceeded by 'DR' - primary or 'DS' - backup)
o Suffix = the file format (always dat, indicating binary data)

```
File name example: Nimbus7_THIRCLDT_1978m1103t232550_o00148_DR6302.TAP
```

2.2 File Format and Structure

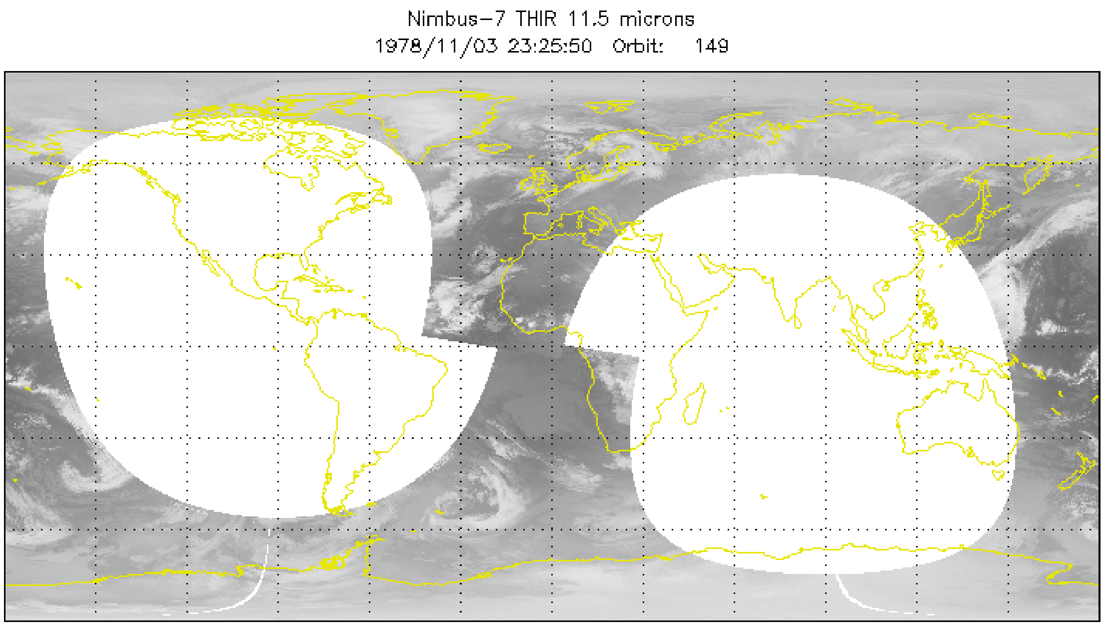
The data are stored as they were originally written in IBM binary (big-endian) record oriented structured files. The files were written on the original 1600 bpi 9-track tapes using a blocked FORTRAN format each with a size of 9288 bytes (2322 words).

There are typically 502 records per data file. The first is a header or documentation record, followed by a set of data records, and the last record is a dummy record to indicate it's the last record in the file. Each data record contains 10 swaths with 92 geolocation points with four 11.5 micron channel pixel samples (368 total), and two 6.7 micron channel pixel samples (184 total). Latitude and longitude information are provided for the first of four 11.5 micron and first of two 6.7 micron pixel samples. The location of the others can be obtained by interpolation. For contents and layout of the documentation and data records, see section 3.1

2.3 Key Science Data Fields

The primary science data fields in this data product is the calibrated radiances in units $W/m^2/sr$ measured at the 11.5 and 6.7 micron channels

Figure 1: Typical data coverage for a Nimbus 7 THIR data file.



3. Data Contents

The granularity of this data collection is a single orbit, approximately 104 minutes.

3.1 Data Records

The Nimbus-7 User's Guide does not describe the layout of the file format, refer to the tables below or see the Nimbus THIR Calibrated Located Data Tape Specification Document No. T344011.

Each data file contains 502 physical records, each of size 9288 bytes. A record identifier is found in the last 12 bits of the first word of each record. The first data record (record id = 10) is the documentation or header record. This is followed by a series of data records (record id = 11) containing. as many as 5000 scans of data. The file is padded out with dummy records (record id = 15) to make the file have 502 records.

Table 3-1-1: Documentation Record (id = 10)

Word	Field Name		Units	Type	Comments
1	Physical Record Number (always 1)		-	12 bits	Bits 20 – 31
	Spare		-	4 bits	Bits 16 – 19
	Record Id (bits 0-5: type, bit 6: last file, bit 7:last record)		-	8 bits	Bits 8 – 15
	Spare		-	8 bits	Bits 0 – 7
2	File Number		-	I*4	
3	Data Orbit Number		-	I*4	
4	Data Orbit Start	Year	-	I*4	
5		Day of Year	-	I*4	
6		Time	msec	I*4	
7	Data Orbit Stop	Year	-	I*4	
8		Day of Year	-	I*4	
9		Time	msec	I*4	

10	Southern Terminator Crossing	Year	-	I*4	
11		Day of Year	-	I*4	
12		Time	msec	I*4	
13	Northern Terminator Crossing	Year	-	I*4	
14		Day of Year	-	I*4	
15		Time	msec	I*4	
16	Longitude of Descending Node		degrees	I*4	Scale factor 1/10
17	Longitude of Ascending Node		degrees	I*4	Scale factor 1/10
18	Ascending Node	Year	-	I*4	
19		Day of Year	-	I*4	
20		Time	msec	I*4	
21	Solar Declination of Ascending Node		degrees	I*4	Scale factor 1/1000
22 | 149	Radiance to Temperature Table For 6.7 Micron Channel		Kelvin	256 x I*2	Scale factor 1/64
150 | 277	Radiance to Temperature Table For 11.5 Micron Channel		Kelvin	256 x I*2	Scale factor 1/64
278 | 2322	Spares		-	8180 bytes	Bits set to zero

Table 3-1-2: Data Record (type 11)

Word	Field Name	Units	Type	Comments
1	Physical Record Number (positive integer > 1)	-	12 bits	Bits 20 – 31
	Spare	-	4 bits	Bits 16 – 19
	Record Id (bits 0-5: type, bit 6: last file, bit 7:last record)	-	8 bits	Bits 8 – 15
	Spare	-	8 bits	Bits 0 – 7
2 | 232	THIR Scan Block #1	-	924 bytes	(see table 3-1-3)
233 | 463	THIR Scan Block #2	-	924 bytes	(see table 3-1-3)
464 | 2311	THIR Scan Blocks #3 through #10	-	924 bytes	(see table 3-1-3)
2312 | 2314	THIR Engineering and Housekeeping Data: a. 3 scan housing temperatures (scale factor 0.2) b. Scan motor temperature (scale factor 0.2) c. Electronics temperature (scale factor 0.2) d. 2 bolometer temperatures (scale factor 0.2) e. 2 average space-level counts f. 2 average housing-level counts g. Spare	°C (Temp) - (Counts)	12 bytes	
2315 | 2322	Spares	-	8 bytes	Bits set to zero

Table 3-1-3: THIR Scan Block

Byte	Field Name	Units	Type	Comments
1 – 2	Time of Nadir View Scan	msec	I*2	
3 – 4	Scan flags		16 bits	
5 | 14	Radiance Block # 1k		10 bytes	(see table 3-1-4)
15 | 24	Radiance Block # 2		10 bytes	(see table 3-1-4)
25 | 924	Radiance Blocks #3 through #92		10 bytes	(see table 3-1-4)

Table 3-1-4: Radiance Block

Byte	Field Name	Units	Type	Comments
1 2	Latitude (9 bit integer part + 7 bit binary fraction)	degrees	I*2	0 – 180 (from south pole)
3 4	Longitude (9 bit integer part + 7 bit binary fraction)	degrees	I*2	0 – 360
5	Radiance #1 at 11.5 microns	$W/m^2/sr$	I*1	Scale factor 1/8
6	Radiance #1 at 6.7 microns	$W/m^2/sr$	I*1	Scale factor 1/64
7	Radiance #2 at 11.5 microns	$W/m^2/sr$	I*1	Scale factor 1/8
8	Radiance #3 at 11.5 microns	$W/m^2/sr$	I*1	Scale factor 1/8
9	Radiance #2 at 6.7 microns	$W/m^2/sr$	I*1	Scale factor 1/64
10	Radiance #4 at 11.5 microns	$W/m^2/sr$	I*1	Scale factor 1/8

Table 3-1-5: Dummy Record (type 15)

Word	Field Name	Units	Type	Comments
1	Physical Record Number (positive integer max 502)	-	12 bits	Bits 20 – 31
	Spare	-	4 bits	Bits 16 – 19
	Record Id (bits 0-5: type, bit 6: last file, bit 7:last record)	-	8 bits	Bits 8 – 15
	Spare	-	8 bits	Bits 0 – 7
2 | 2322	Spares	-	9284 bytes	Bits set to zero

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	Satellite 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 Reverb

The GES DISC provides basic temporal and advanced (event) searches through the EOSDIS Reverb data search and download interface:

http://reverb.echo.nasa.gov

Reverb allows users the ability to search on keywords, spatial region, and time period on datasets archived and various data centers. It offers various download options that suit users with different preferences and different levels of technical skills. To search for the THIR data enter GES_DISC_THIRN7L1CLDT_V001 into the keyword field.

5.2 FTP

The Nimbus data products are available for users to download directly using anonymous FTP:

ftp://acdisc.gsfc.nasa.gov/data/s4pa/Nimbus7_THIR_Level1/THIRN7L1CLDT.001/

The data are organized in directories by year with subdirectories by day of year. README, User's Guide and other documentation are located under the doc directory.

6. More Information

6.1 Web Resources

For other Nimbus data products, please see the GES DISC's Nimbus heritage data web page at:
http://disc.gsfc.nasa.gov/nimbus/

To search for other related data, please visit NASA's Global Change Master Directory at:
http://gcmd.nasa.gov.

6.2 Point of Contact

Name:	GES DISC Help Desk
URL:	http://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

6.3 References

"The Nimbus-7 User's Guide - Section 9", NASA Goddard Space Flight Center, Aug. 1978, Pages 247-263

7. Appendices

Acknowledgements

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

THIR: Temperature Humidity Infrared Radiometer

L1: Level-1 Data

NASA: National Aeronautics and Space Administration

Reverb: ECHO's Next Generation Metadata and Service Discovery Tool

QA: Quality Assessment

UT: Universal Time

Image Files

The THIRN7IM data product contains scanned positives of photofacsimile 70mm film strips from the Nimbus-7 THIR instrument. The images contain a full orbit from the 6.7 micron cloud cover and another with the 11.5 micron channel of the Earth's surface temperature. Each orbital swath picture is gridded with geographic coordinates and covers a distance approximately from the north pole to the south pole. About 7 days of images are archived into a TAR file. The THIRN7IM images can be ordered online using the Reverb/ECHO tool at http://reverb.echo.nasa.gov/, and then enter GES_DISC_THIRN7IM_V001 into the keyword field.

The THIRN7IM images are available for a few select days from 1984/04/14 through 1984/06/06 (orbits 27626 through 28375), and can be used as a reference to the THIRN7L1CLDT digital data. The image files can be viewed with any application that supports the JPEG 2000 format.

FORTRAN Code

```
C-----------------------------------------------------------------------
C ^NAME: READ_CLDT
C
C ^DESCRIPTION:
C     This program opens and reads a Nimbus-7 THIR level-1 CLDT data
C     file and prints the contents of the file to the screen. Data files
C     consist of a header record, followed by about 500 data records,
C     followed by a terminating dummy record.  See the Nimbus-7 User's
C     Guide, Section 9 for a description of THIR.
C
C ^MAJOR VARIABLES:
C     FNAME  - name of input file
C     BLOCK  - buffer for data block typically has three data records
C     BUFF   - buffer for holding temporary 4-byte word
C     WORD   - integer 4-byte word
C     IBLKSZ - size of block in bytes
C     IOS    - I/O status number
C
C ^NOTES:
C     Compile: gfortran -o READ_CLDT.EXE READ_CLDT.FOR
C
C ^ORGANIZATION: NASA/GSFC, Code 610.2
C
C ^AUTHOR:  James Johnson
C
C ^ADDRESS: james.johnson@nasa.gov
C
C ^CREATED: May 20, 2015
C-----------------------------------------------------------------------

      CHARACTER          FNAME*1024
      CHARACTER          BLOCK(9288)   ! Buffer = 9288 bytes
      CHARACTER          BUFF(4)       ! Buffer to hold 4-byte word
      INTEGER*4          WORD          ! 4-byte word
      INTEGER*4          IBLKSZ        ! Block size header
      EQUIVALENCE        (BUFF, WORD)

C     Get the name of the input data file to read
      PRINT *, 'Enter the name of the input file:'
      READ (5,'(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 (block number) and IOFF (byte offset in file)
      N=1
      IOFF=0

C     Loop through the file reading all blocks of data
      DO

C       Read the first 4-byte word or block size header
        DO I=1,4
```

```
          READ (1, REC=IOFF+I, IOSTAT=IOS, ERR=90) BUFF(I)
        END DO
        IBLKSZ = WORD
        IOFF=IOFF+I-1

        IF (IBLKSZ .EQ. 0) THEN
C         PRINT '("WARNING: END-OF-TAPE MARK")'
          GOTO 20
        ENDIF

C       Next read the block of data
        DO I=1,IBLKSZ
          READ (1, REC=IOFF+I, IOSTAT=IOS) BLOCK(I)
          IF (IOS .NE. 0) THEN
            PRINT '("ERROR: BLOCK ",I4,X,I4,", IOSTAT: ",I6)', N,I-1,IOS
            IBLKSZ = I-1
            GOTO 10
          ENDIF
        END DO

C       Check the record type. This is byte 2 (3 unswapped) of first
C       4-byte word. Value 10 = header, 11 = data, 15 = dummy
   10   ITYPE = IAND(ICHAR(BLOCK(3)), B'00111111')
        IF (ITYPE .EQ. 10) THEN
          CALL PRHREC(BLOCK, IBLKSZ, N)
        ELSE IF (ITYPE .EQ. 11) THEN
          CALL PRDREC(BLOCK, IBLKSZ, N)
        ELSE IF (ITYPE .EQ. 15) THEN
          CALL PRXREC(BLOCK, IBLKSZ, N)
        ELSE
          PRINT '("Unknown record type: ", I3)', ITYPE
        ENDIF
        IOFF=IOFF+I-1

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

        N=N+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 ^SUBROUTINE: PRHREC
C
C     This Subroutine will Print the Documentation/Header Record
C-----------------------------------------------------------------------

      SUBROUTINE PRHREC(WRDARR, IBLKSZ, N)

      INTEGER*4           WRDARR(2322)  ! Word Array
      CHARACTER           BUFF*4        ! Temporary data buffer
      INTEGER*4           I4BUF         ! 4-byte integer buffer
      INTEGER*2           I2BUF(2)      ! 2-byte integer buffer
      INTEGER*2           IRECNO        ! Physical Record Number
      INTEGER*1           IRECID        ! Record Id
      INTEGER*4           FILENO,       ! File number
     &                    ORBNUM,       ! Orbit number
     &                    STIME(3),     ! Orbit Start Year/Day/Time (msec)
     &                    ETIME(3),     ! Orbit Stop Year/Day/Time (msec)
     &                    STXTIM(3),    ! So. Term. Crossing Year/Day/Time
     &                    NTXTIM(3),    ! No. Term. Crossing Year/Day/Time
     &                    LONDSC,       ! Longitude Descending Node
     &                    LONASC,       ! Longitude Ascending Node
     &                    TIMASC(3),    ! Ascending Node Year/Day/Time (msec)
     &                    SOLDEC        ! Solar Declination of Ascending Node
      INTEGER*2           R2T115(256),  ! Radiance Temperature Table 11.5 um
     &                    R2T67(256)    ! Radiance Temperature Table 6.7 um
      CHARACTER           SWPBYT*4      ! Function for swapping bytes
      EQUIVALENCE         (BUFF, I4BUF, I2BUF)

      PRINT '("***************")'

C     Physical Record Number and Record Id
      I4BUF = WRDARR(1)
      BUFF = SWPBYT(BUFF(1:4), 4)
      IRECNO = ISHFT(I2BUF(2),-4)
      PRINT '("RECNO  =",X,I6)', IRECNO
      IRECID = ICHAR(BUFF(2:2))
      PRINT '("RECID  =",X,I3)', IRECID

C     File Number
      I4BUF = WRDARR(2)
      BUFF = SWPBYT(BUFF(1:4), 4)
      FILENO = I4BUF
      PRINT '("FILENO =",X,I11)', FILENO

C     Data Orbit Number
      I4BUF = WRDARR(3)
      BUFF = SWPBYT(BUFF(1:4), 4)
      ORBNUM = I4BUF
      PRINT '("ORBNUM =",X,I11)', ORBNUM

C     Data Orbit Start Time: Year/Day of Year/Time of Day (msec)
      I4BUF = WRDARR(4)
      BUFF = SWPBYT(BUFF(1:4), 4)
      STIME(1) = I4BUF
      I4BUF = WRDARR(5)
      BUFF = SWPBYT(BUFF(1:4), 4)
```

```
      STIME(2) = I4BUF
      I4BUF = WRDARR(6)
      BUFF = SWPBYT(BUFF(1:4), 4)
      STIME(3) = I4BUF
      PRINT '("STIME  =",3(X,I11))', STIME

C     Data Orbit Stop Time: Year/Day of Year/Time of Day (msec)
      I4BUF = WRDARR(7)
      BUFF = SWPBYT(BUFF(1:4), 4)
      ETIME(1) = I4BUF
      I4BUF = WRDARR(8)
      BUFF = SWPBYT(BUFF(1:4), 4)
      ETIME(2) = I4BUF
      I4BUF = WRDARR(9)
      BUFF = SWPBYT(BUFF(1:4), 4)
      ETIME(3) = I4BUF
      PRINT '("ETIME  =",3(X,I11))', ETIME

C     Southern Terminator Crossing Time: Year/Day of Year/Time of Day (msec)
      I4BUF = WRDARR(10)
      BUFF = SWPBYT(BUFF(1:4), 4)
      STXTIM(1) = I4BUF
      I4BUF = WRDARR(11)
      BUFF = SWPBYT(BUFF(1:4), 4)
      STXTIM(2) = I4BUF
      I4BUF = WRDARR(12)
      BUFF = SWPBYT(BUFF(1:4), 4)
      STXTIM(3) = I4BUF
      PRINT '("STXTIM =",3(X,I11))', STXTIM

C     Northern Terminator Crossing Time: Year/Day of Year/Time of Day (msec)
      I4BUF = WRDARR(13)
      BUFF = SWPBYT(BUFF(1:4), 4)
      NTXTIM(1) = I4BUF
      I4BUF = WRDARR(14)
      BUFF = SWPBYT(BUFF(1:4), 4)
      NTXTIM(2) = I4BUF
      I4BUF = WRDARR(15)
      BUFF = SWPBYT(BUFF(1:4), 4)
      NTXTIM(3) = I4BUF
      PRINT '("NTXTIM =",3(X,I11))', NTXTIM

C     Longitude of Descending Node
      I4BUF = WRDARR(16)
      BUFF = SWPBYT(BUFF(1:4), 4)
      LONDSC = I4BUF
      PRINT '("LONDSC =",X,G12.6)', LONDSC/10.

C     Longitude of Ascending Node
      I4BUF = WRDARR(17)
      BUFF = SWPBYT(BUFF(1:4), 4)
      LONASC = I4BUF
      PRINT '("LONASC =",X,G12.6)', LONASC/10.

C     Time of Ascending Node: Year/Day of Year/Time of Day (msec)
      I4BUF = WRDARR(18)
      BUFF = SWPBYT(BUFF(1:4), 4)
```

```
      TIMASC(1) = I4BUF
      I4BUF = WRDARR(19)
      BUFF = SWPBYT(BUFF(1:4), 4)
      TIMASC(2) = I4BUF
      I4BUF = WRDARR(20)
      BUFF = SWPBYT(BUFF(1:4), 4)
      TIMASC(3) = I4BUF
      PRINT '("TIMASC =",3(X,I11))', TIMASC

C     Solar Declination at Ascending Node
      I4BUF = WRDARR(21)
      BUFF = SWPBYT(BUFF(1:4), 4)
      SOLDEC = I4BUF
      PRINT '("SOLDEC =",X,G12.6)', SOLDEC/1000.

C     Radiance to Temperature Table for 6.7 micron channel
      DO 10 I=22,149
        I4BUF = WRDARR(I)
        BUFF = SWPBYT(BUFF(1:4), 4)
        R2T67(2*(I-22)+1) = I2BUF(2)
        R2T67(2*(I-22)+2) = I2BUF(1)
   10 CONTINUE
      PRINT '("R2T67  =",8(X,F7.3),/,(8X,8(X,F7.3)))', R2T67/64.

C     Radiance to Temperature Table for 11.5 micron channel
      DO 20 I=150,277
        I4BUF = WRDARR(I)
        BUFF = SWPBYT(BUFF(1:4), 4)
        R2T115(2*(I-150)+1) = I2BUF(2)
        R2T115(2*(I-150)+2) = I2BUF(1)
   20 CONTINUE
      PRINT '("R2T115 =",8(X,F7.3),/,(8X,8(X,F7.3)))', R2T115/64.

      RETURN
      END
```

```
C----------------------------------------------------------------------
C ^SUBROUTINE: PRDREC
C
C     This Subroutine will Print the Documentation/Header Record
C----------------------------------------------------------------------

      SUBROUTINE PRDREC(WRDARR, IBLKSZ, N)

      INTEGER*4          WRDARR(2322)  ! Word Array
      INTEGER*4          SCNBLK(231)   ! SCAN Block Buffer
      CHARACTER          BUFF*4        ! Temporary data buffer
      INTEGER*4          I4BUF         ! 4-byte integer buffer
      INTEGER*2          I2BUF(2)      ! 2-byte integer buffer
      INTEGER*2          IRECNO        ! Physical Record Number
      INTEGER*1          IRECID        ! Record Id
      CHARACTER          THOUSE(3)     ! 3 Scan Housing Temperatures
      CHARACTER          TMOTOR        ! Scan Motor Temperature
      CHARACTER          TELECT        ! Electronics Tempeature
      CHARACTER          TBOLOM(2)     ! 2 Bolometer Tempeatures
      CHARACTER          TAVGSC(2)     ! 2 Average Space-Level Counts
      CHARACTER          TAVGHC(2)     ! 2 Average Housing-Level Counts
      CHARACTER          SWPBYT*4      ! Function for swapping bytes
      EQUIVALENCE        (BUFF, I4BUF, I2BUF)

      PRINT '("***************")'

C     Physical Record Number and Record Id
      I4BUF = WRDARR(1)
      BUFF = SWPBYT(BUFF(1:4), 4)
      IRECNO = ISHFT(I2BUF(2),-4)
      PRINT '("RECNO  =",X,I6)', IRECNO
      IRECID = ICHAR(BUFF(2:2))
      PRINT '("RECID  =",X,I3)', IRECID

C     Loop through 10 THIR Scan Blocks of 231 Words / 924 Bytes
      DO 10 I=1,10
      PRINT '("--------------")'
        SCNBLK = WRDARR((I-1)*231+2:(I)*231+1)
        CALL PRSCAN(SCNBLK)
   10 CONTINUE
      PRINT '("--------------")'

C     Words 2312 - 2314 hold the Engineering and Housekeeping Info
      I4BUF = WRDARR(2312)
      THOUSE(1) = BUFF(1:1)
      THOUSE(2) = BUFF(2:2)
      THOUSE(3) = BUFF(3:3)
      PRINT '("THOUSE =",3(X,G12.6))', ICHAR(THOUSE)*0.2
      TMOTOR = BUFF(4:4)
      PRINT '("TMOTOR =",X,G12.6)', ICHAR(TMOTOR)*0.2

      I4BUF = WRDARR(2313)
      TELECT = BUFF(1:1)
      PRINT '("TELECT =",X,G12.6)', ICHAR(TELECT)*0.2
      TBOLOM(1) = BUFF(2:2)
      TBOLOM(2) = BUFF(3:3)
      PRINT '("TBOLOM =",2(X,G12.6))', ICHAR(TBOLOM)*0.2
```

```
      TAVGSC(1) = BUFF(4:4)

      I4BUF = WRDARR(2314)
      TAVGSC(2) = BUFF(1:1)
      PRINT '("TAVGSC =",2(X,G12.6))', ICHAR(TAVGSC)*1.0
      TAVGHC(1) = BUFF(2:2)
      TAVGHC(2) = BUFF(3:3)
      PRINT '("TAVGHC =",2(X,G12.6))', ICHAR(TAVGHC)*1.0

      RETURN
      END


C-----------------------------------------------------------------------
C ^SUBROUTINE: PRSCAN
C
C     This Subroutine will Print the THIR SCAN Blocks
C-----------------------------------------------------------------------

      SUBROUTINE PRSCAN(BYTARR)

      CHARACTER          BYTARR*924    ! Byte Array
      CHARACTER          BUFF*4        ! Temporary data buffer
      CHARACTER          THIRWD*10     ! 80-bit THIR Word
      INTEGER*4          I4BUF         ! 4-byte integer buffer
      INTEGER*2          I2BUF(2)      ! 2-byte integer buffer
      INTEGER*2          IRECNO        ! Physical Record Number
      INTEGER*1          IRECID        ! Record Id
      INTEGER*2          TIME          ! Time of Nadir Scan (1/4 sec from
start)
      INTEGER*2          FLAGS         ! Data Flags for THIR Scan
      INTEGER*2          ILON,         ! Packed Longitude
     &                   ILAT          ! Packed Latitude
      CHARACTER          RADARR(6)     ! Radiance samples
      REAL*4             RAD115(4)     ! Radiance at 11.5 microns (W/m2/sr)
      REAL*4             RAD67(2)      ! Radiance at  6.7 microns (W/m2/sr)
      REAL*4             R4CLDT        ! Function for converting to real
      CHARACTER          SWPBYT*4      ! Function for swapping bytes
      EQUIVALENCE        (BUFF, I4BUF, I2BUF)

C     First Word in each THIR Scan Block contains Time (1/4 secs) and Data
Flags

      BUFF(1:4) = BYTARR(1:4)
      BUFF = SWPBYT(BUFF(1:4), 4)
      TIME = I2BUF(2)
      PRINT '("TIME   =",X,G12.6)', TIME/4.
      FLAGS = I2BUF(1)
      PRINT '("FLAGS  =",X,B16.16)', FLAGS

C     Each THIR Scan Block contains 92 80-bit Scan Words of Data
      DO 100 J=1,92

        THIRWD(1:10) = BYTARR((J-1)*10+5:J*10+4)

        BUFF(1:4) = THIRWD(1:4)
        BUFF = SWPBYT(BUFF(1:4), 4)
```

```
      ILAT = I2BUF(2)
      IF (ILAT .NE. -1) THEN
        PRINT '("LAT    =",X,G12.6,I6)', R4CLDT(ILAT) - 90.
      ELSE
        PRINT '("LAT    =",X,G12.6,I6)', R4CLDT(ILAT)
      ENDIF
      ILON = I2BUF(1)
      PRINT '("LON    =",X,G12.6)', R4CLDT(ILON)

      DO 10 I=1,6
        RADARR(I) = THIRWD(I+4:I+4)
   10 CONTINUE
c     PRINT '("RADARR =",6(X,I3))', ICHAR(RADARR)

      IF (ICHAR(RADARR(1)) .NE. 255) THEN
        RAD115(1) = ICHAR(RADARR(1))/8.
      ELSE
        RAD115(1) = -1.0
      ENDIF
      IF (ICHAR(RADARR(2)) .NE. 255) THEN
        RAD67(1) = ICHAR(RADARR(2))/64.
      ELSE
        RAD67(1) = -1.0
      ENDIF
      IF (ICHAR(RADARR(3)) .NE. 255) THEN
        RAD115(2) = ICHAR(RADARR(3))/8.
      ELSE
        RAD115(2) = -1.0
      ENDIF
      IF (ICHAR(RADARR(4)) .NE. 255) THEN
        RAD115(3) = ICHAR(RADARR(4))/8.
      ELSE
        RAD115(3) = -1.0
      ENDIF
      IF (ICHAR(RADARR(5)) .NE. 255) THEN
        RAD67(2) = ICHAR(RADARR(5))/64.
      ELSE
        RAD67(2) = -1.0
      ENDIF
      IF (ICHAR(RADARR(6)) .NE. 255) THEN
        RAD115(4) = ICHAR(RADARR(6))/8.
      ELSE
        RAD115(4) = -1.0
      ENDIF

      PRINT '("RAD115 =",4(X,G12.6))', RAD115
      PRINT '("RAD67  =",2(X,G12.6))', RAD67

  100 CONTINUE

    RETURN
    END
```

```
C-----------------------------------------------------------------------
C ^SUBROUTINE: PRXREC
C
C     This Subroutine will Print the Dummy Record
C-----------------------------------------------------------------------

      SUBROUTINE PRXREC(WRDARR, IBLKSZ, N)

      INTEGER*4          WRDARR(2322)  ! Word Array
      CHARACTER          BUFF*4        ! Temporary data buffer
      INTEGER*4          I4BUF         ! 4-byte integer buffer
      INTEGER*2          I2BUF(2)      ! 2-byte integer buffer
      INTEGER*2          IRECNO        ! Physical Record Number
      INTEGER*1          IRECID        ! Record Id
      CHARACTER          SWPBYT*4      ! Function for swapping bytes
      EQUIVALENCE        (BUFF, I4BUF, I2BUF)

      PRINT '("***************")'

C     Physical Record Number and Record Id
      I4BUF = WRDARR(1)
      BUFF = SWPBYT(BUFF(1:4), 4)
      IRECNO = ISHFT(I2BUF(2),-4)
      PRINT '("RECNO  =",X,I6)', IRECNO
      IRECID = ICHAR(BUFF(2:2))
      PRINT '("RECID  =",X,I3)', IRECID

C     The rest of the bytes are spares and set to zero.

      RETURN
      END
```

```
C-----------------------------------------------------------------------
C ^FUNCTION: SWPBYT
C
C     This function will swap the bytes of a data element
C-----------------------------------------------------------------------

      CHARACTER*4 FUNCTION SWPBYT(DATBUF, NBYTES)

      CHARACTER          DATBUF*4       ! Input data buffer
      CHARACTER          TEMP*4         ! Output swapped buffer

      DO 10 K=1,NBYTES
        SWPBYT(K:K) = DATBUF(NBYTES-K+1:NBYTES-K+1)
   10 CONTINUE

      RETURN
      END


C-----------------------------------------------------------------------
C ^FUNCTION: R4CLDT
C
C     This function will convert an input short to a float
C-----------------------------------------------------------------------

      FUNCTION R4CLDT(ISHORT)

      INTEGER*2      ISHORT             ! 16-bit short integer
      INTEGER*4      I                  ! Integer part
      REAL*4         F                  ! Fraction part

      I = ISHFT(ISHORT,-7)
      F = IAND(ISHORT,B'0000000001111111')/2.**7
      R4CLDT = I + F

      RETURN
      END
```