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

README Document for the Synchronous Meteorological Satellite (SMS) 2 Visible Infrared Spin-Scan Radiometer (VISSR) Level 1 Experimenter History Tape (EHT) Data

VISSRSMS2L1EHT

Last Revised 05/14/2020

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	05/14/2020
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***
05/14/2020	Original	James E. Johnson

Table of Contents

1. Introduction

This document provides basic information on using the Synchronous Meteorological Satellite (SMS) 2 Visible Infrared Spin-Scan Radiometer (VISSR) Level-1 Experimenter History Tape (EHT) product.

1.1 Data Product Description

The SMS 2 Visible Infrared Spin-Scan Radiometer (VISSR) Level-1 Experimenter History Tape data product typically contains a segment of the Earth with radiances that were measured in the visible (0.55 to 0.70 micrometer) and IR (10.5 to 12.6 micrometer) wavelengths with a spatial resolution of 0.9 and 8 km, respectively. Files also include time, geolocation, orbit, attitude, and telemetry information. A data file is structured with a header, followed by an IR scan line and then 8 visible scan lines. Visible scans are at full resolution of 15288 pixels and a file will contain several hundred scan lines. IR scans are at 3822 pixels and up to a hundred scan lines. A full scan of the Earth was made every 20 minutes.

Data for this product are only available for the following 5 days: 1975/02/17, 1975/04/24, 1975/04/25, 1975/05/06, and 1975/08/28. The SMS-2 satellite was initially parked over the equator at longitude 105W on February 22, 1975 viewing the hemisphere below the satellite. It was later moved to longitude 135W on Dec 18, 1975 for increased coverage of the Western Pacific. The VISSR experiment was operated by the NOAA National Environmental Satellite Data and Information Service (NESDIS), as well as scientists from NASA Goddard Space Flight Center.

This product was previously available from the NSSDC with the identifier ESAD-00039 (old ID 75-011A-04A).

1.1.1 The Visible Infrared Spin-Scan Radiometer

The Visible Infrared Spin-Scan Radiometer (VISSR) flown on SMS 2 was basically identical to that flown on SMS 1, and provided day and night observations of cloudcover and earth/cloud radiance temperature measurements from a synchronous, spin-stabilized, geostationary satellite for use in operational weather analysis and forecasting. The two-channel instrument was able to take both full and partial pictures of the earth's disk. The infrared channel (10.5 to 12.6 micrometers) and the visible channel (0.55 to 0.70 micrometer) used a common optics system. Incoming radiation was received by an elliptically shaped scan mirror and collected by a Ritchey-Chretien optical system. The scan mirror was set at a nominal angle of 45 deg to the VISSR optical axis, which was aligned parallel to the spin axis of the spacecraft. The spinning motion of the spacecraft (approximately 100 rpm) provided a west-to-east scan motion when the spin axis of the spacecraft was oriented parallel to the earth's axis. The latitudinal scan was accomplished by sequentially tilting the scanning mirror north to south at the completion of each spin. A full picture took 18.2 min to complete and about 2

min to retrace. During each scan, the field of view on the earth was swept by a linear array of eight visible-spectrum detectors, each with a ground resolution of 0.9 km at zero nadir angle. The infrared portion of the detector measured radiance temperatures between 180 and 315 deg K with a proposed sensitivity between 0.4 and 1.4 deg K.

1.1.2 The SMS 2 Satellite

The SMS 2 satellite was successfully launched on February 6, 1975. SMS 2 was the second in a series of geosynchronous meteorological satellites developed by NASA and operated by NOAA, which would evolve into NOAA's GOES system. The spacecraft included three instruments: (1) a visible infrared spin-scan radiometer (VISSR) which provided high-quality day/night cloudcover data and made radiance temperatures of the earth/atmosphere system, (2) a meteorological data collection and transmission system (DCS) which relayed processed data from central weather facilities to small automatic picture transmission (APT) equipped regional stations and collected and retransmitted data from remotely located earth-based platforms, and (3) a space environmental monitor (SEM) which measured proton, electron, and solar X-ray fluxes and magnetic fields.

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

Parking Station	**Longitude**	**Date**
Launch		02/06/1975
Original	105° W	02/22/1975
Final	135° W	12/19/1975

Geostationary circular orbit at 35,000 km

Inclination of 1.9 degrees

1.2 Algorithm Background

The SMS 2 VISSR 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 copied to 9-track tapes for archival. Further information on the VISSR instrument and data processing can be found in the "VISSR Data Processing Plan for SMS/GOES Satellites" and the "The GOES/SMS User's Guide".

1.3 Data Disclaimer

The data should be used with care and one should first read both the “VISSR Data Processing Plan for SMS/GOES Satellites” and the “GOES/SMS User’s Guide”. Users should cite this data product in their research.

2. Data Organization

Data from the SMS 2 Visible Infrared Spin-Scan Radiometer (VISSR) EHT product are only available for the following 5 days: 1975/02/17, 1975/04/24, 1975/04/25, 1975/05/06, and 1975/08/28. Each file typically contains a few minutes of data on the order of a few 100 visible scans and up to a 100 IR scans.

2.1 File Naming Convention

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

<Platform>-<Instrument>_<Level>-<Product>_<DateTime>_<TapeNumber>-<TapeFile>.<Suffix>

where:
o Platform = name of the platform or satellite (SMS2)
o Instrument = name of the instrument and product (VISSR)
o Level = process level (L1)
o Product = product name (EHT)
o Date = Data start date and time in UTC in format <YYYY>m<MMDD>t<hhmmss> where

1. YYYY = 4 digit year (1975)
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 TapeNumber = 4 or 5 digit number of tape (preceded by ‘DR’ - primary or ‘DS’ or ‘DD’ - backup)
o TapeFile = 1 digit number of file on the tape
o Suffix = the file format (always TAP, indicating tape binary data)

```
File name example: SMS2-VISSR_L1-EHT_1975m0217t180053_DR5094-5.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 originally written on 9-track 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 data files. Each data file on the tape contains a label record of size 132 bytes, followed by a repeating set of 9 scan records (a single IR scan and 8 visible scans). The scan records are 11871 bytes on both SSEC and Sandstone tapes, with the data packed into 2638 x 36-bit words. Each data file typically contains a few minutes of data (18 minutes for a full global image). For the contents and layout of the data, see Appendix B of the “VISSR Data Processing Plan for SMS/GOES Satellites”.

During data recovery a total of 373 data files were archived out of 1139 data files, the others were either duplicates, near-duplicates, less complete, or corrupted files. The data files came from tapes designated by a prefix 'DR' (primary from Sandstone) or 'DS' (backup from SSEC) and a five digit number. Of the tapes, 66 were from SSEC, and 65 were from Sandstone. Caution should be taken as some data records contain corrupted records. The GMT time stamp is typically correct, though it was discovered that on some files the time had to be obtained from the alternate corrected time stamp in the IR or Visible documentation section, or from the tape header. Each file is unique, although two files, SMS2-VISSR_L1-EHT_1975m0217t191158_DR5090-10.TAP and SMS2-VISSR_L1-EHT_1975m0217t191158_DR5093-15.TAP, should both have142 IR records but the former is missing record 55 in the latter, and the latter is missing record 124 in the former. Also, the two files SMS2-VISSR_L1-EHT_1975m0424t221305_DR5099-7.TAP and SMS2-VISSR_L1-EHT_1975m0424t221305_DR5099-7.TAP each have some better scans in one rather than the other, so both are archived. Almost all files archived are from Sandstone tapes with the exception of the following from SSEC:

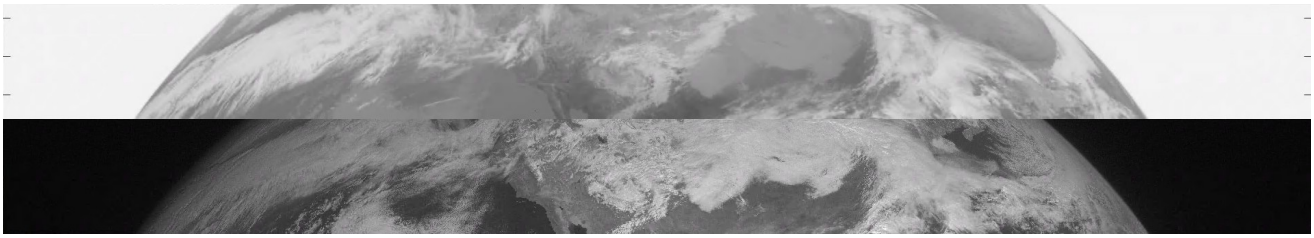
1. SMS2-VISSR_L1-EHT_1975m0217t184326_DS5088-2.TAP
2. SMS2-VISSR_L1-EHT_1975m0217t180053_DD23502-1.TAP
3. SMS2-VISSR_L1-EHT_1975m0217t180256_DD23501-1.TAP
4. SMS2-VISSR_L1-EHT_1975m0217t180327_DD23501-2.TAP
5. SMS2-VISSR_L1-EHT_1975m0217t180432_DD23501-3.TAP
6. SMS2-VISSR_L1-EHT_1975m0217t180654_DD23499-1.TAP

All of the recovered SMS 2 VISSR EHT files are archived at the GES DISC.

2.3 Key Science Data Fields

The primary science data fields in this data product are the VISSR calibrated IR and visible radiances.

Figure 1: A typical SMS 2 VISSR Level 1 EHT data file showing an infrared scene (cool = white, warm = black) at top, and visible scene at bottom from February 17, 1975 at 18:33:26 to 18:34:56 UTC.



3. Data Contents

The granularity of this data product is typically around 20 minutes.

3.1 Data Records

The file format is described in Appendix B of the "VISSR Data Processing Plan for SMS/GOES Satellites" document.

The original tape files each included a tape header file. These were then followed by a set of up to 18 data files. As part of the recovery, the GES DISC has extracted and archived the data files from the tapes. Each data file includes a header label, followed by either a set of one IR and eight visible scans, or just a set of IR scans. The original data were written on IBM machines using 36-bit words. There were two types of tapes received. The first from University of Wisconsin SSEC contains record blocks preceded by 8 BCD characters indicating the record block size, during extraction the 36-bit words were packed into 4½ bytes. The second from Sandstone contains record blocks preceded by a 4-byte word with the record block size, during extraction each byte contains 6 bits of a 36-bit word.

Table 3-1-1: Label Record (132 EBCDIC characters if 9-track, or 132 BCD characters if 7-track)

Character	Description	Size	Comments
1 – 7	International Code	7	7501101 for SMS 2
9 – 14	Date of Recording (YYMMDD)	6	
16 – 18	Station Code	3	
20	Analog Tape Deck Identification	1	
22 – 27	Tape Number	6	
29 – 32	Start Time of Analog Tape (HHMM)	4	
34 – 37	Stop Time of Analog Tape (HHMM)	4	
39 – 44	Date of Digitization (YYMMDD)	6	
46	Picture Section Number	1	
48	Buffer Tape Reel Number	1	
50	Buffer Tape File Number	1	
52	Buffer Tape Section File Number	1	
54 – 55	Pass Two Line Identification	2	
57	Pre-Edit Tape Reel Number	1	
59	Pre-Edit Tape File Number	1	
61	Pre-Edit Tape Selection File Number	1	Section file is relative to first file

63	Master Data Tape Reel Number	1	
65	Master Data Tape File Number	1	
66 – 67	Master Data Tape Section File Number	2	Section file is relative to first file
69 – 71	Start Day of Year (3 digit)	3	
72 – 77	Start Time (HHMMSS)	6	
79 – 84	Stop Time (HHMMSS)	6	
86 – 89	Elapsed Time (MMSS)	4	
91	Data Mode	1	A = Visible Mode A B = Visible Mode B G = IR Grid I = IR Image K = IR Calibration
93	IR Data Only (Y or N)	1	
95 – 98	Initial Image Line Number	4	
100 – 103	Final Image Line Number	4	
105 – 109	DECOM Run Number	5	
111 – 112	DECOM Reel Number	2	
114	DECOM Reel File Number	1	
115 – 116	DECOM Section File Number	2	Section file is relative to first file
122	O/A/TM Data Present (Y or N)	1	
124	GMT Present From Time Track (Y, N or D)	1	Y = Valid GMT N = Flywheel GMT, D = from Documentation Time
126 – 130	Creation Date (YMMDD)	5	
132	Tape Identification	1	E = Experimenter History Tape

Table 3-1-2: IR Data Record (1108 36-bit words in 11871 bytes)

Word	Subfield (Bits)	Contents
1 – 32		IR Documentation (129 x 9-Bit Bytes) *see table 3-2-1 below*
32 – 987		IR Scan Line Data (3822 x 9-Bit Bytes)
988	35 – 18	
	17 – 0	Zero Fill
989 – 1628		First Telemetry Data Block, When Available (640 x 36-Bit Words) – no definition
1629 – 1643		First Orbit/Attitude Data Block, When Available (15 x 36-Bit Words) – no definition*
1644 – 1658		Zero Fill
1659 – 2298		Second Telemetry Data Block, When Available (640 x 36-Bit Words) – no definition
2299 – 2313		Second Orbit/Attitude Data Block, When Available (15 x 36-Bit Words) – no definition*
2314 – 2635		Zero Fill
2636	35 – 12	
	11 – 0	F1 Pre-Edit Tape Data Flags (12 Bits)
2637	35 – 24	F2 MDT and EHT Data Flags (12 Bits)
	23 – 12	F3 Pre-Edit Tape Image Line Number (12 Bits)
	11 – 0	F4 Corrected Scan Line Number (12 Bits)
2638		GMT or Flywheel Time (Bits 35 – 27 are Day of Year; Bits 26 – 0 are Milliseconds)

* Word 1 is Time (Bits 35 – 27 are Day of Year; Bits 26 – 0 are Milliseconds)

Table 3-1-3: IR Documentation

Bytes	Contents
1	Retrace[1] – one indicates scanner retrace
2	Spacecraft Name
3	Unused
4	Frame Code[1] – one indicates picture transmission
5	Change Code[1] – one indicates start of picture if frame code is one or end of picture if frame code is zero
6	Step Code[1] – one indicates normal line transmission; zero indicates that this is not to be used to expose film and facsimile recorder line is not to be incremented (stepped).
7	Line Delay – this number (1 – 8) denotes delay to be introduced by user; expressed in bit in intervals.
8	IR Selection – 000000001 = IR1, 000000010 = IR2, 000000100 = AVG
9	Gray Scale Status[1] – one indicates gray scale information retransmission
10	Direct Transmission Mode[1] – one indicates 28 Mb/sec; zero indicates 14 Mb/sec
11	Image Line Number – BCD value split into 2 characters/word 2 most significant BCD characters
12	2 least significant BCD characters
13	Mode Code – A = 001100100, B = 01011001, C = 010110010, D = 010001111

14	Beta Count *MSB* 0 8 MSB *LSB*
15	0 8 MSB
16	0 8 LSB
17	Grid/No Grid[1] – Zero indicates no grid information
18	Sync Error *MSB* *LSB* 0 8 MSB
19	0 7 LSB 0
20	Bit Error Count - *MSB* *LSB* 0 8 MSB
21	0 5 LSB 000
22	Setup Error[1] – one indicates setup error
23 - 24	Computer Error Message *MSB* 00000001 *LSB* 00000010 ⋮ ⋮ 10000000
25	Scan Count – Two most significant BCD characters
26	Two least significant BCD characters
27	Year – 2 MSD in BCD characters
28	Year – 2 LSD in BCD characters
29	Day of Year – 2 MSD in BCD characters
30	Day of Year – 2 MSD in BCD characters
31	Hour – 2 BCD characters)
32	Minute – 2 BCD characters
33	Second – 2 BCD characters)
34	Millisecond × 10 – 2 BCD characters
35	Black Enable1 – one indicates annotation transmission
36	Mode C-Calibrate[1] – one indicates C-Cal is not used; otherwise: *MSB* V_1 00000001 V_2 00000010 V_3 00000100 V_4 00001000 V_5 00010000 V_6 00100000 V_7 01000000 V_8 10000000
37	Bit/Frame Sync Lock *MSB* *LSB* 00000001X Bit Lock 1 loss 00000010X Frame Lock X = 1 for any 00000100X Bit Freq Lock of lock
38	Limited Scan Mode Indicator[1] – one indicates limited scan mode

39	Sample Control Mode *(LSB)* IR – 2 PT IR – 1 PT IR – EMP VIS – 4 PT VIS – 2 PT VIS – 1 PT VIS - EMT
40	Visible Channel Connection *(LSB)* V1 V2 V3 Coding for each channel is V4 as follows: V5 0 = normal V6 1 = patched input is used V7 V8
41	Scan Direction[1] – one indicates normal North-South direction (may not be used)
42	Bi-phase Modulator On/Off[1] – one indicates modulator is on
43	Unused
44	PLL Error Light[1] – one indicates error condition
45	Test Data *MSB LSB* 000000000 Normal 000000001 Local 000000010 Remote 000000100 Comp Gen IR
46	Data Randomization[1] – one indicates on
47	Sun Pulse Select[1] – one indicates digital, zero indicates analog
48	Ness Mode Select *MSB LSB* 000000000 4 x 4 IR 000000001 Max SV 000000010 4 x 2 IR
49	Limited Scan Command Encoder Enable[1] – one indicates on
50	Digital Sun Pulse – 8 LSB's contain the digital sun pulse
51	Bit Error Light[1] – one indicates on
52	Mean IR Difference
53	RMS IR Difference
54 – 55	Correction Table ID
56	Left Horizon Point – 8 MSB
57	8 LSB
58	Right Horizon Point – 8 MSB
59	8 LSB
60	Equatorial Scan Count – 8 MSB
61	8 LSB

62 – 79	Unused			
80	Telemetry Code Word *MSB* *LSB* 0 A B C D E F G H A and B Unused; always zero C = D = 0 Indicates frame sync search; no valid telemetry, three or more bad sync patterns since lock C = 0, D = 1 Indicates frame sync in check1; one good sync since search or two bad syncs since lock C = 1, D = 0 Indicates frame sync in check2; two good sync since search or one bad sync since lock C = D = 1 Indicates frame sync in lock; three or more good sync patterns found EFG = 0-3 Telemetry quarter-frame number 4 First output without telemetry 5 Second output without telemetry 6 Third output without telemetry 7 Fourth or greater output without telemetry H Parity bit, exclusive or of B, D and F			
81 – 98	One quarter of telemetry frame; 16 9-bit words packed into the 8 LSBs of each 9-bit IR doc word			
99	O/A block number (1 – 10) One quarter of an orbit/attitude data block; contents are as follows[2]:			
	First Quarter	Second Quarter	Third Quarter	Fourth Quarter
100	0000001	00000010	00000011	00000100
101 – 104	O/A Word 2	O/A Word 8	O/A Word 14	B_Φ
105 – 108	O/A Word 3	O/A Word 9	O/A Word 15	$\dot{B}*2^{30}$
109 – 112	O/A Word 4	O/A Word 10	O/A Word 16	τ_Φ (HHMMSS) *2
113 – 116	O/A Word 5	O/A Word 11	O/A Word 17	O/A Word 20
117 – 120	O/A Word 6	O/A Word 12	O/A Word 18	O/A Word 21
121 – 124	O/A Word 7	O/A Word 13	O/A Word 19	O/A Word 3
125 – 127	Unused			
128	Longitudinal Parity Check (ODD)			

Note 1 All but the last bit in each code is identical, e.g., 000000001 (zero) or 111111110 (one)

Note 2 Regarding O/A data blocks, however words 100 to 124:

a) If no O/A is present, word 100 will be zeros

b) Descriptions of each O/A word are contained in Table A-2

c) B_Φ, $\dot{B}$, τ_Φ are the coefficients of the equation for beta count, words 14-16.
The units of $\dot{B}$ are counts per milliseconds,
and the second bit of $\dot{B}$ is set to 1 to indicate a negative value.

IR Doc Words 80 – 128 appear to be mostly unused in EHT

Table 3-1-4: VIS Mode A Record (1108 36-bit words in 11871 bytes if SSEC tape)

Word	Subfield (Bits)	Contents
1		Visible Documentation (7 x 6-Bit Bytes)
2	35 – 30	
	29 – 0	Unused
3 – 40		
41	35 – 18	Corrected Equatorial Scan Count
	17 – 0	Corrected Year (2 Decimal Digit)
42		Corrected Time (Bits 35 – 27 are Day of Year; Bits 26 – 0 are Milliseconds)
43 – 85		Unused
86	35 – 24	
	23 – 0	Visible Sensor Line Data (15,288 x 6-Bit Bytes)
87 – 2633		
2634	35 – 24	
	23 – 0	Zero Fill
2635		
2636	35 – 12	
	11 – 0	F1 Pre-Edit Tape Data Flags (12 Bits)
2637	35 – 24	F2 MDT and EHT Data Flags (12 Bits)
	23 – 12	F3 Pre-Edit Tape Image Line Number (12 Bits)
	11 – 0	F4 Corrected Scan Line Number (12 Bits)
2638		GMT or Flywheel Time (Bits 35 – 27 are Day of Year; Bits 26 – 0 are Milliseconds)

Note: there are no VIS Mode B files in this collection

Table 3-1-5: Visible Documentation

Bytes	Contents
1 – 3	Sector Code* – (1 – 8) uses three words; each word represents a zero or one state. The most significant word is first. Mode A, the sectors (following IR) have numbers 000, 001, 010, 011, 100, 101, 110, 111; in mode B, the sector numbers are 000, 001, 010, and 011.
4	
	Frame Code* – one indicates picture transmission
5	Change Code* – one indicates start of picture if frame code is one or end of picture if frame code is zero
6	Step Code* – one indicates normal line transmission; zero indicates that this is not to be used to expose film and facsimile recorder line is not to be incremented (stepped).
7	Line Offset – this is a three bit word from the line offset logic inserted into the last three bit positions with 0'x inserted into the first three positions, i.e., (000XXX).

* All but the last bit in each code is identical, e.g., 000001 (zero) or 111110 (one)

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

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

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.
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 points that represent the satellite coverage. Each point is identified by its latitude and longitude pair.
Gpolygon: PointLongitude	Longitudes of the polygon points that represent the satellite coverage. Each point is identified by its latitude and longitude pair.
Recovering_Institute_Tape_ID	The Tape ID used by the data recovery institute
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 36-bit words, thus users will have to reconstruct the data values accordingly

A sample FORTRAN program is included in the Appendix section which will read in the data records.

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/VISSRSMS2L1EHT_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/SMS/VISSRSMS2L1EHT.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 GOES/SMS User's Guide", ed. Corbell R., Callahan, C, Kotsch, W., NOAA National Environmental Satellite Services, 1976

"VISSR Data Processing Plan for SMS/GOES Satellites" by McCowan, P. L., NASA Goddard Space Flight Center, September 1977

7. Appendices

Acknowledgments

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

Acronyms

EOS: Earth Observing System

EHT: Experimenter History Tapes

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

SMS: Synchronous Meteorological Satellite

VISSR: Visible Infrared Spin-Scan Radiometer

FORTRAN Code

```
C-----------------------------------------------------------------------
C ^NAME: READ_SMSEHT
C   This program reads SMS VISSR Experimenter History Tape (EHT) files.
C
C   The SMS VISSR files contain a label record (size 132 bytes),
C   followed by a series of data records (typically 11871 bytes).
C   The data records can be a set of one IR scan and 8 visible scans
C   or just a set of IR scans. Data records contain the IR or visible
C   radiances, as well as time, and other ancillary values. This program
C   will print the contents of each data record.
C
C   Please note the record size markers can vary. For University of
C   Wisconsin SSEC recovered files it is an 8 BCD character value
C   preceding the record. For Sandstone recovered files it is a 4-byte
C   integer at the beginning and end of each record. This program checks
C   for both types.
C
C ^MAJOR VARIABLES:
C     FNAME  - name of input file
C     IBLKSZ - size of record block in bytes
C     BUFF   - buffer for data record
C     WD36   - array for 36-bit words
C     IOS    - I/O status number
C
C ^NOTES:
C     Compile: gfortran -o READ_SMSEHT.EXE READ_SMSEHT.FOR
C
C ^AUTHOR:  James Johnson (James.Johnson@nasa.gov), NASA GES DISC
C
C ^HISTORY: Feb 11, 2020 - first version for SMS1
C           May 13, 2020 - tested on both SMS1 and SMS2 files
C-----------------------------------------------------------------------
      PROGRAM READ_SMSEHT
      CHARACTER FNAME*1024                  ! Filename
      CHARACTER BUFF(15828)                 ! Buffer for data record block
C                                           ! 15828 bytes (36 bits/6 bytes)
C                                           ! 11871 bytes (36 bits/4.5 bytes)
      INTEGER*8 WD36(2638)                  ! Array for 36-bit words
      INTEGER*4 IBLKSZ                      ! Size of records
      INTEGER*4 IWORD                       ! 4-byte word
      INTEGER*4 CHKSMS                      ! Check if IR/VIS record
      CHARACTER SSEC*8                      ! Buffer to hold 8-byte SSEC 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 = ",A160)', 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
      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)
          SSEC(I:I) = TEMP(I)
        END DO
        IOFF=IOFF+(I-1)

        M=0
        DO I=1,4
          IF (ICHAR(TEMP(I)).GE.48.AND.ICHAR(TEMP(I)).LE.57) M=M+1
        END DO
        IF (M.EQ.4) THEN
          DO I=1,4
            READ (1, REC=IOFF+I, IOSTAT=IOS, ERR=90) TEMP(I)
            SSEC(I+4:I+4) = TEMP(I)                        ! UW SSEC tape file
          END DO
          IOFF=IOFF+(I-1)
          READ (SSEC,'(I8)') IBLKSZ
        ELSE
          IBLKSZ = IWORD                                   ! Sandstone tape file
        END IF

C End-of-File (EOF) mark, continue
        IF (IBLKSZ .EQ. 0) GOTO 90

C Next read the block of data
        DO I=1,15828
          BUFF(I)=CHAR(0)                                  ! Clear the buffer
        END DO
        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 90
          END IF
        END DO
        IOFF=IOFF+(I-1)
        N=N+1

        IF (N.EQ.1.AND.IBLKSZ.LE.132) THEN                 ! Label Record
          CALL PRLABL(IBLKSZ,BUFF)
        ELSE                                               ! Data Scan Record
          CALL BLK2WD(BUFF,IBLKSZ,WD36)
          IF (CHKSMS(WD36(1)) .EQ. 0) THEN
            CALL PRIREC(WD36)
          ELSE
            CALL PRVREC(WD36)
          END IF
        END IF
```

```
        IF (M.EQ.0) THEN                                    ! End block size word
          DO I=1,4                                          ! Sandstone tape file
            READ (1, REC=IOFF+I, IOSTAT=IOS, ERR=90) TEMP(I)
          END DO
          IOFF=IOFF+(I-1)
        END IF
      END DO

C Close the input file
   90 CLOSE(1)
      STOP

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

C-----------------------------------------------------------------------
C     This Subroutine unpacks 36-bit words into 64-bit long long integer
C-----------------------------------------------------------------------
      SUBROUTINE BLK2WD(BUFF, IBLKSZ, WD36)
      CHARACTER BUFF(15828)                          ! Buffer for record block
      REAL*4 NPACK                                   ! 36-bit packing method
      INTEGER*8 W36BIT                               ! Use 64-bit for 36-bit int
      INTEGER*8 WD36(2638)                           ! Array of 64-bit words

C     Correct record blocks that aren't the correct size
      IF (IBLKSZ.EQ.15828) THEN                      ! Sandstone file 15828bytes
        NPACK = 6.0
      ELSE                                           ! SSEC/Sandstone 11861bytes
        NPACK = 4.5
C       IF (IBLKSZ.LT.11871) THEN                    ! Record is short,add bytes
C         Add bytes, but how?
C       END IF
C       IF (IBLKSZ.GT.11871) THEN                    ! Record is long,chop bytes
C         Remove bytes, but how?
C       END IF
      END IF

C     Store 36-bit words into 64-bit double long IEEE integers
      DO N=1,2638
        IF (NPACK.EQ.4.5) THEN
          WD36(N) =
     &    W36BIT(BUFF(INT((N-1)*NPACK)+1:INT(N*NPACK+0.5)), N)
        ELSE
          WD36(N) =
     &    W36BIT(BUFF(INT((N-1)*NPACK)+1:INT(N*NPACK)), 0)
        END IF
      END DO

      RETURN
      END
```

```
C-----------------------------------------------------------------------
C     This Function will convert data into a 36-bit word
C-----------------------------------------------------------------------
      INTEGER*8 FUNCTION W36BIT(BUFF,N)
      CHARACTER BUFF(8)
      CHARACTER TEMP(8)
      INTEGER*8 WD36
      EQUIVALENCE (TEMP,WD36)

      IF (N.NE.0) THEN
        DO I=1,8
          IF (I.LE.5) THEN
            TEMP(8-I-2) = BUFF(I)                          ! swap the byte order
          ELSE
            TEMP(I) = CHAR(0)
          END IF
        END DO
        IF (MOD(N,2).EQ.0) THEN
          WD36 = IAND(WD36,'FFFFFFFFF'Z)
        ELSE
          WD36 = ISHFT(WD36,-4)
        END IF
      ELSE
        WD36 = 0
        DO I=1,8
          WD36 = ISHFT(WD36, 6)                            ! shift left 6 bits
          WD36 = IOR(WD36,IAND(ICHAR(BUFF(I)),'3F'Z))      ! remove 2 MSBs
        END DO
      END IF

      RETURN
      END
```

```
C-----------------------------------------------------------------------
C     This Subroutine will print the Label Record
C-----------------------------------------------------------------------
      SUBROUTINE PRLABL(IBLKSZ,BUFF)
      CHARACTER BUFF(132)                           ! Buffer for record block
      CHARACTER ASC*132                             ! Buffer for ASCII label

      ICHK = 0                                      ! EBCDIC encoded text
      IF (ICHAR(BUFF(1)).LT.64) ICHK = 1            ! BCD encoded text

      DO I = 1,IBLKSZ
        IF (ICHK.EQ.1) THEN
          ASC(I:I) = CHAR(IBCD(INT(IAND(ICHAR(BUFF(I)),'3F'Z))))
        ELSE
          ASC(I:I) = CHAR(IEBC(ICHAR(BUFF(I))))
        END IF
      END DO

      PRINT '("IntlCode   = ",A7)', ASC(1:7)
      PRINT '("DateRecd   = ",A6)', ASC(9:14)
      PRINT '("Station    = ",A3)', ASC(16:18)
      PRINT '("TapeDeckId = ",A1)', ASC(20:20)
      PRINT '("TapeNo     = ",A6)', ASC(22:27)
      PRINT '("TapeStart  = ",A4)', ASC(29:32)
      PRINT '("TapeStop   = ",A4)', ASC(34:37)
      PRINT '("DigitzDate = ",A6)', ASC(39:44)
      PRINT '("PicSectNo  = ",A1)', ASC(46:46)
      PRINT '("BuffReelNo = ",A1)', ASC(48:48)
      PRINT '("BuffFileNo = ",A1)', ASC(50:50)
      PRINT '("BuffSectNo = ",A1)', ASC(52:52)
      PRINT '("Pass2Line  = ",A2)', ASC(54:55)
      PRINT '("PrEdReelNo = ",A1)', ASC(57:57)
      PRINT '("PrEdFileNo = ",A1)', ASC(59:59)
      PRINT '("PrEdSectNo = ",A1)', ASC(61:61)
      PRINT '("MdtReelNo  = ",A1)', ASC(63:63)
      PRINT '("MdtFileNo  = ",A1)', ASC(65:65)
      PRINT '("MdtSectNo  = ",A2)', ASC(66:67)
      PRINT '("StartDay   = ",A3)', ASC(69:71)
      PRINT '("StartTime  = ",A6)', ASC(72:77)
      PRINT '("StopTime   = ",A6)', ASC(79:84)
      PRINT '("ElapTime   = ",A4)', ASC(86:89)
      PRINT '("DataMode   = ",A1)', ASC(91:91)
      PRINT '("IROnlyFlag = ",A1)', ASC(93:93)
      PRINT '("InitLineNo = ",A4)', ASC(95:98)
      PRINT '("LastLineNo = ",A4)', ASC(100:103)
      PRINT '("DecomRun   = ",A5)', ASC(105:109)
      PRINT '("DecomReel  = ",A2)', ASC(111:112)
      PRINT '("DecomFile  = ",A1)', ASC(114:114)
      PRINT '("DecomSect  = ",A2)', ASC(115:116)
      PRINT '("OATmFlag   = ",A1)', ASC(122:122)
      PRINT '("GMTFlag    = ",A1)', ASC(124:124)
      PRINT '("CreateDate = ",A5)', ASC(126:130)
      PRINT '("TapeId     = ",A1)', ASC(132:132)
      PRINT '("============")'

      RETURN
      END
```

```
C-----------------------------------------------------------------------
C     This Subroutine will print the IR Record
C-----------------------------------------------------------------------
      SUBROUTINE PRIREC(WD36)
      INTEGER*8 WD36(2638)                          ! IR record (36-bit words)
      INTEGER*2 IRDOC(128)                          ! IR doc (9-bit words)
      INTEGER*2 IRDATA(3822)                        ! IR data (9-bit words)

C     IR Documentation
      DO I=1,32
        IRDOC(4*(I-1)+1) = IAND(ISHFT(WD36(I),-27),'1FF'Z)
        IRDOC(4*(I-1)+2) = IAND(ISHFT(WD36(I),-18),'1FF'Z)
        IRDOC(4*(I-1)+3) = IAND(ISHFT(WD36(I), -9),'1FF'Z)
        IRDOC(4*(I-1)+4) = IAND(ISHFT(WD36(I),  0),'1FF'Z)
      END DO
      PRINT '("Retrace    =",X,I3)', IRDOC(1)
      PRINT '("Spacecraft =",X,I3)', IRDOC(2)
C     PRINT '("Unused     =",X,I3)', IRDOC(3)
      PRINT '("FrameCode  =",X,I3)', IRDOC(4)
      PRINT '("ChangeCode =",X,I3)', IRDOC(5)
      PRINT '("StepCode   =",X,I3)', IRDOC(6)
      PRINT '("LineDelay  =",X,I3)', IRDOC(7)
      PRINT '("IRSelect   =",X,I3)', IRDOC(8)
      PRINT '("GrySclStat =",X,I3)', IRDOC(9)
      PRINT '("XmitMode   =",X,I3)', IRDOC(10)
      PRINT '("ImgLineNo  =",4(X,I3))',
     &  IAND(ISHFT(IRDOC(11),-4),'0F'Z),IAND(IRDOC(11),'0F'Z),
     &  IAND(ISHFT(IRDOC(12),-4),'0F'Z),IAND(IRDOC(12),'0F'Z)
      PRINT '("ModeCount  =",X,I3)', IRDOC(13)
      PRINT '("BetaCount  =",3(X,I3))', IRDOC(14:16)
      PRINT '("GridNoGrid =",X,I3)', IRDOC(17)
      PRINT '("SyncError  =",2(X,I3))', IRDOC(18:19)
      PRINT '("BitErrCnt  =",2(X,I3))', IRDOC(20:21)
      PRINT '("SetupError =",X,I3)', IRDOC(22)
      PRINT '("CompError  =",2(X,I3))', IRDOC(23:24)
      PRINT '("ScanCount  =",4(X,I3))',
     &  IAND(ISHFT(IRDOC(25),-4),'0F'Z),IAND(IRDOC(25),'0F'Z),
     &  IAND(ISHFT(IRDOC(26),-4),'0F'Z),IAND(IRDOC(26),'0F'Z)
      PRINT '("CorrYear   =",4(X,I3))',
     &  IAND(ISHFT(IRDOC(27),-4),'0F'Z),IAND(IRDOC(27),'0F'Z),
     &  IAND(ISHFT(IRDOC(28),-4),'0F'Z),IAND(IRDOC(28),'0F'Z)
      PRINT '("CorrDay    =",4(X,I3))',
     &  IAND(ISHFT(IRDOC(29),-4),'0F'Z),IAND(IRDOC(29),'0F'Z),
     &  IAND(ISHFT(IRDOC(30),-4),'0F'Z),IAND(IRDOC(30),'0F'Z)
      PRINT '("CorrHour   =",2(X,I3))',
     &  IAND(ISHFT(IRDOC(31),-4),'0F'Z),IAND(IRDOC(31),'0F'Z)
      PRINT '("CorrMin    =",2(X,I3))',
     &  IAND(ISHFT(IRDOC(32),-4),'0F'Z),IAND(IRDOC(32),'0F'Z)
      PRINT '("CorrSec    =",2(X,I3))',
     &  IAND(ISHFT(IRDOC(33),-4),'0F'Z),IAND(IRDOC(33),'0F'Z)
      PRINT '("CorrMSec   =",2(X,I3))',
     &  IAND(ISHFT(IRDOC(34),-4),'0F'Z),IAND(IRDOC(34),'0F'Z)
      PRINT '("BlkEnable  =",X,I3)', IRDOC(35)
      PRINT '("ModeC_Cal  =",X,I3)', IRDOC(36)
      PRINT '("BFSyncLock =",X,I3)', IRDOC(37)
      PRINT '("LmtScnMode =",X,I3)', IRDOC(38)
```

```
      PRINT '("SmpScnMode =",X,I3)', IRDOC(39)
      PRINT '("VisChCnnct =",X,I3)', IRDOC(40)
      PRINT '("ScanDir    =",X,I3)', IRDOC(41)
      PRINT '("BiPhaseMod =",X,I3)', IRDOC(42)
C     PRINT '("Unused     =",X,I3)', IRDOC(43)
      PRINT '("PLLError   =",X,I3)', IRDOC(44)
      PRINT '("TestData   =",X,I3)', IRDOC(45)
      PRINT '("DataRandom =",X,I3)', IRDOC(46)
      PRINT '("SunPulsSel =",X,I3)', IRDOC(47)
      PRINT '("NessMode   =",X,I3)', IRDOC(48)
      PRINT '("LmtScnCmd  =",X,I3)', IRDOC(49)
      PRINT '("DigSunPuls =",X,I3)', IRDOC(50)
      PRINT '("BitErrLght =",X,I3)', IRDOC(51)
      PRINT '("MeanIRDiff =",X,I3)', IRDOC(52)
      PRINT '("RMSIRDiff  =",X,I3)', IRDOC(53)
      PRINT '("CorrTblID  =",2(X,I3))',IRDOC(54:55)
      PRINT '("LfHorizPt  =",X,I7)',ISHFT(IRDOC(56),8)+IRDOC(57)
      PRINT '("RtHorizPt  =",X,I7)',ISHFT(IRDOC(58),8)+IRDOC(59)
      PRINT '("EqScanCnt  =",X,I7)',ISHFT(IRDOC(60),8)+IRDOC(61)
C     PRINT '("Unused     =",2(X,I3))', IRDOC(62:79)
      PRINT '("TelemCode  =",X,I3)', IRDOC(80)
      PRINT '("QtrTelem   =",18(X,I3))', IRDOC(81:98)
      PRINT '("OABlkNum   =",X,I3)', IRDOC(99)
      PRINT '("QtrNum     =",X,I3)', IRDOC(100)
      PRINT '("OABlock    =",24(X,I3))', IRDOC(101:124)
C     PRINT '("Unused     =",3(X,I3))', IRDOC(125:127)
      PRINT '("LonParChk  =",X,I3)', IRDOC(128)

C     IR Data Values
      DO I=33,987
        IRDATA(4*(I-33)+1) = IAND(ISHFT(WD36(I),-27),'1FF'Z)
        IRDATA(4*(I-33)+2) = IAND(ISHFT(WD36(I),-18),'1FF'Z)
        IRDATA(4*(I-33)+3) = IAND(ISHFT(WD36(I), -9),'1FF'Z)
        IRDATA(4*(I-33)+4) = IAND(ISHFT(WD36(I),  0),'1FF'Z)
      END DO
      IRDATA(4*(I-33)+1) = IAND(ISHFT(WD36(I),-27),'1FF'Z)    ! for word 988 use
      IRDATA(4*(I-33)+2) = IAND(ISHFT(WD36(I),-18),'1FF'Z)    ! just bits 35-18
      PRINT '("IR Data    =",/,20(X,I3))', IRDATA

C     First Telemtry Data Block
      PRINT '("TelemBlk1  =",/,6(X,I11))', WD36(989:1628)
C     First Orbit/Attitude Data Block
      PRINT '("OrbAttBlk1 =",/,6(X,I11))', WD36(1629:1643)

C     Second Telemtry Data Block
      PRINT '("TelemBlk2  =",/,6(X,I11))', WD36(1659:2298)
C     Second Orbit/Attitude Data Block
      PRINT '("OrbAttBlk2 =",/,6(X,I11))', WD36(2299:2313)

C     Flags F1: Preedit Tape Data,      F2: Preedit Tape Data
C           F3: Preedit Tape Image Line No., F4: Corrected Scan Line No(MDT/EHT)
      PRINT '("FlagF1     =",X,I4)', IAND(ISHFT(WD36(2636), -0),'FFF'Z)
      PRINT '("FlagF2     =",X,I4)', IAND(ISHFT(WD36(2637),-24),'FFF'Z)
      PRINT '("FlagF3     =",X,I4)', IAND(ISHFT(WD36(2637),-12),'FFF'Z)
      PRINT '("FlagF4     =",X,I4)', IAND(ISHFT(WD36(2637), -0),'FFF'Z)

C     GMT or Flywheel Time (Day + Milliseconds)
```

```
      PRINT '("GMTDay     =",X,I4)', ISHFT(WD36(2638),-27)
      PRINT '("GMTMSec    =",X,I9)', IAND(WD36(2638),'7FFFFFF'Z)
      PRINT '("============")'

      RETURN
      END

C-----------------------------------------------------------------------
C     This Subroutine will print the Vis A Record
C-----------------------------------------------------------------------
      SUBROUTINE PRVREC(WD36)
      INTEGER*8 WD36(2638)                          ! VIS record (36-bit words)
      INTEGER*1 VISDOC(7)                           ! VIS doc (6-bit words)
      INTEGER*4 VISCOR(4)                           ! VIS corrected values
      INTEGER*1 VISDAT(15288)                       ! VIS data (6-bit words)

C     Visible Documentation
      DO I=1,2
        IF (I.EQ.1) THEN
          VISDOC(1) = IAND(ISHFT(WD36(I),-30),'3F'Z)
          VISDOC(2) = IAND(ISHFT(WD36(I),-24),'3F'Z)
          VISDOC(3) = IAND(ISHFT(WD36(I),-18),'3F'Z)
          VISDOC(4) = IAND(ISHFT(WD36(I),-12),'3F'Z)
          VISDOC(5) = IAND(ISHFT(WD36(I), -6),'3F'Z)
          VISDOC(6) = IAND(ISHFT(WD36(I), -0),'3F'Z)
        ELSE
          VISDOC(7) = IAND(ISHFT(WD36(I),-30),'3F'Z)
        END IF
      END DO
      PRINT '("SectorCode =",3(X,I2))', VISDOC(1:3)
      PRINT '("FrameCode  =",X,I2)', VISDOC(4)
      PRINT '("ChangeCode =",X,I2)', VISDOC(5)
      PRINT '("StepCode   =",X,I2)', VISDOC(6)
      PRINT '("LineOffset =",X,I2)', VISDOC(7)

C     Unused - WD36(2),bits:29-0 and WD36(3:40)

C     Corrected Equatorial Scan Count, Year and Time (Day + Milliseconds)
      VISCOR(1) = ISHFT(WD36(41),-18)
      VISCOR(2) = IAND(WD36(41),'3FFFF'Z)
      VISCOR(3) = ISHFT(WD36(42),-27)
      VISCOR(4) = IAND(WD36(42),'7FFFFFF'Z)
      PRINT '("CorrEqScan =",X,I4)', VISCOR(1)
      PRINT '("CorrYear   =",X,I4)', VISCOR(2)
      PRINT '("CorrDay    =",X,I4)', VISCOR(3)
      PRINT '("CorrMSec   =",X,I9)', VISCOR(4)

C     Unused - WD36(43-85) and WD36(86),bits:35-24

C     VIS Data Values
      DO I=86,2634
        IF (I.EQ.86) THEN
          VISDAT(6*(I-86)+1) = IAND(ISHFT(WD36(I),-18),'3F'Z) ! For word 85 use
          VISDAT(6*(I-86)+2) = IAND(ISHFT(WD36(I),-12),'3F'Z) ! just bits 23-0
          VISDAT(6*(I-86)+3) = IAND(ISHFT(WD36(I), -6),'3F'Z)
          VISDAT(6*(I-86)+4) = IAND(ISHFT(WD36(I),  0),'3F'Z)
        ELSE IF (I.LT.2634) THEN
```

```
          VISDAT(6*(I-86)-1) = IAND(ISHFT(WD36(I),-30),'3F'Z)
          VISDAT(6*(I-86)+0) = IAND(ISHFT(WD36(I),-24),'3F'Z)
          VISDAT(6*(I-86)+1) = IAND(ISHFT(WD36(I),-18),'3F'Z)
          VISDAT(6*(I-86)+2) = IAND(ISHFT(WD36(I),-12),'3F'Z)
          VISDAT(6*(I-86)+3) = IAND(ISHFT(WD36(I), -6),'3F'Z)
          VISDAT(6*(I-86)+4) = IAND(ISHFT(WD36(I),  0),'3F'Z)
        ELSE
          VISDAT(6*(I-86)-1) = IAND(ISHFT(WD36(I),-30),'3F'Z) ! for word 988 use
          VISDAT(6*(I-86)+0) = IAND(ISHFT(WD36(I),-24),'3F'Z) ! just bits 35-24
        END IF
      END DO
      PRINT '("VIS Data   =",/,20(X,I3))', VISDAT

C     Zero Fill - WD36(2634),bits:23-0, WD36(2635) and WD36(2636),bits:35-12

C     Flags F1: Preedit Tape Data,      F2: Preedit Tape Data
C           F3: Preedit Tape Image Line No., F4: Corrected Scan Line No(MDT/EHT)
      PRINT '("FlagF1     =",X,I4)', IAND(ISHFT(WD36(2636), -0),'FFF'Z)
      PRINT '("FlagF2     =",X,I4)', IAND(ISHFT(WD36(2637),-24),'FFF'Z)
      PRINT '("FlagF3     =",X,I4)', IAND(ISHFT(WD36(2637),-12),'FFF'Z)
      PRINT '("FlagF4     =",X,I4)', IAND(ISHFT(WD36(2637), -0),'FFF'Z)

C     GMT or Flywheel Time (Day + Milliseconds)
      PRINT '("GMTDay     =",X,I4)', ISHFT(WD36(2638),-27)
      PRINT '("GMTMSec    =",X,I9)', IAND(WD36(2638),'7FFFFFF'Z)
      PRINT '("============")'

      RETURN
      END
```

```
C-----------------------------------------------------------------------
C     This Function checks if IR Record or VIS Record
C-----------------------------------------------------------------------
      FUNCTION CHKSMS(WD36)
      INTEGER*8 WD36                                        ! 36-bit word
      INTEGER*2 BIT9(4)                                     ! 4 x 9-bits in word
      INTEGER*1 BIT6(6)                                     ! 6 x 6-bits in word
      INTEGER*4 CHKSMS                                      ! check value

C     IR uses 4 x 9-bits in a word
      BIT9(1) = IAND(ISHFT(WD36,-27),'1FF'Z)                ! Retrace
      BIT9(2) = IAND(ISHFT(WD36,-18),'1FF'Z)                ! Spacecraft
      BIT9(3) = IAND(ISHFT(WD36, -9),'1FF'Z)                ! 0 (unused) if IR
      BIT9(4) = IAND(ISHFT(WD36,  0),'1FF'Z)                ! Frame Code
C     VIS uses 6 x 6-bits in a word
      BIT6(1) = IAND(ISHFT(WD36,-30),'3F'Z)                 ! Section Code
      BIT6(2) = IAND(ISHFT(WD36,-24),'3F'Z)                 ! "       "
      BIT6(3) = IAND(ISHFT(WD36,-18),'3F'Z)                 ! "       "
      BIT6(4) = IAND(ISHFT(WD36,-12),'3F'Z)                 ! Frame Code
      BIT6(5) = IAND(ISHFT(WD36, -6),'3F'Z)                 ! Change Code
      BIT6(6) = IAND(ISHFT(WD36,  0),'3F'Z)                 ! Step Code

      CHKSMS=-1
      IF (BIT9(3).EQ.0) THEN
        CHKSMS=0
      ELSE
        IF (BIT6(1).EQ.1) THEN
          IF (BIT6(2).EQ.1) THEN
            IF (BIT6(3).EQ.1) THEN
              CHKSMS=5
            ELSE IF (BIT6(3).EQ.62) THEN
              CHKSMS=6
            ENDIF
          ELSE IF (BIT6(2).EQ.62) THEN
            IF (BIT6(3).EQ.1) THEN
              CHKSMS=7
            ELSE IF (BIT6(3).EQ.62) THEN
              CHKSMS=8
            ENDIF
          ENDIF
        ELSE IF (BIT6(1).NE.999) THEN
          IF (BIT6(2).EQ.1) THEN
            IF (BIT6(3).EQ.1) THEN
              CHKSMS=1
            ELSE IF (BIT6(3).EQ.62) THEN
              CHKSMS=2
            ENDIF
          ELSE IF (BIT6(2).EQ.62) THEN
            IF (BIT6(3).EQ.1) THEN
              CHKSMS=3
            ELSE IF (BIT6(3).EQ.62) THEN
              CHKSMS=4
            ENDIF
          ENDIF
        END IF
      END IF
```

```
      RETURN
      END

C-----------------------------------------------------------------------
C     This Function will return BCD to ASCII character index
C-----------------------------------------------------------------------
      FUNCTION IBCD(I)
      CHARACTER BCDTBL(64)

      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_
C                  _0  _1  _2  _3  _4  _5  _6  _7

      IBCD = ICHAR(BCDTBL(I+1))

      RETURN
      END

C-----------------------------------------------------------------------
C     This Function returns EBCDIC to ASCII character index
C-----------------------------------------------------------------------
      FUNCTION IEBC(I)
      INTEGER EBCTBL(256)

      DATA EBCTBL /
     & 000,001,002,003,255,009,255,127,255,255,255,011,012,013,014,015, ! 0_
     & 016,017,018,019,255,133,008,255,024,025,255,255,028,029,030,031, ! 1_
     & 255,255,255,255,255,010,023,027,255,255,255,255,255,005,006,007, ! 2_
     & 255,255,022,255,255,255,255,004,255,255,255,255,020,021,255,026, ! 3_
     & 032,255,255,255,255,255,255,255,255,255,162,046,060,040,043,124, ! 4_
     & 038,255,255,255,255,255,255,255,255,255,033,036,042,041,059,172, ! 5_
     & 045,047,255,255,255,255,255,255,255,255,166,044,037,095,062,063, ! 6_
     & 255,255,255,255,255,255,255,255,255,096,058,035,064,039,061,034, ! 7_
     & 255,097,098,099,100,101,102,103,104,105,255,255,255,255,255,177, ! 8_
     & 255,106,107,108,109,110,111,112,113,114,255,255,255,255,255,255, ! 9_
     & 255,126,115,116,117,118,119,120,121,122,255,255,255,255,255,255, ! a_
     & 094,255,255,255,255,255,255,255,255,255,091,093,255,255,255,255, ! b_
     & 123,065,066,067,068,069,070,071,072,073,255,255,255,255,255,255, ! c_
     & 125,074,075,076,077,078,079,080,081,082,255,255,255,255,255,255, ! d_
     & 092,255,083,084,085,086,087,088,089,090,255,255,255,255,255,255, ! e_
     & 048,049,050,051,052,053,054,055,056,057,255,255,255,255,255,159/ ! f_
C       _0  _1  _2  _3  _4  _5  _6  _7  _8  _9  _A  _B  _C  _D  _E  _F

      IEBC = EBCTBL(I+1)

      RETURN
      END
```