Project

MDPS Systems

Multi-Satellite Data Processing Systems

Title

Transcribed Data Format on DLT

Version 6.8

	NAME	DATE	SIGNATURE
Prepared by:	ACS Team	10/12/2009	
Verified by:	S. Folco (ACS)	10/12/2009	
Approved by:	A. Vollono (ACS)	10/12/2009	
Authorized by:	R. Medri (ACS)	10/12/2009	

DOCUMENT STATUS SHEET

ISSUE	DATE	PAGES or SECTIONS	OBSERVATIONS
1.0	30/06/1998		LANDSAT 7 DLT TRANSCRIPTION Version
2.0	04/09/1998		added J-ERS 1 SAR DLT TRANSCRIPTION Version
3.0	15/09/1998		added ERS SAR DLT TRANSCRIPTION Version
3.1	18/09/1999	32,33	modified J-ERS 1 SAR : Format Description correction
4.0	25/09/1998		added SPOT 1,2,3,4 DLT TRANSCRIPTION Version
4.1	01/02/1999		modified SPOT 1,2,3 : Format Description correction
5.0	01/08/1999	repaginated completely	added XSAR (SRL 1 & 2 and SRTM mission)
5.1	01/01/2000		changed XSAR format of the Storage Area
5.2	01/06/2000		changed XSAR ephemeris file
6.0	10/08/2001	repaginated completely	Added EOSAM/EOSPM (Modis) DLT archiving format
6.1	24/06/2004	5, 73	Added Annex with DLT WILMA stored on disk
6.2	26/11/2004	19, 48, 68	Corrected Transcription Date format in Pass Header, User Header and Statistic files
6.3	26/01/2005	73	Extended WILMA on DISK naming convention to include pass start time
6.4	27/04/2005	70,71,72	Updated Station Name Codes table
6.5	27/07/2005	43,44	Corrected swath size for EOS(AM/PM) Modis
6.6	14/03/2006	9, 15, 23, 26, 29, 33, 34, 35, 53, 70, 71	• Corrected number of files in Transcription area for EOSPM case • Corrected SatelliteTime meaning for Landsat (Etm, Tm, Mss) • Added JERS 1 VNIR format specification
6.7	02/04/2007	51, 55, 56, 57, 58, 59	• Corrected field offset description in User Header structure • Corrected SPOT card structures description (fields offset and meaning)
6.8	10/12/2009	27	• Corrected the Sensor Scan Data description

Table of Contents

1. INTRODUCTION

1.1 SCOPE

A.C.S. stations for Ingestion, Archiving and Cataloguing of satellite data are based on Digital Linear Tape (DLT) units for long term archiving.
This document describes the data format of the transcribed data on the DLT cassette. The description is valid for Transcription Systems developed by A.C.S. for LANDSAT satellites (TM, MSS and ETM+ sensors), J-ERS 1 (SAR, VNIR sensors), ERS 1-2 (SAR sensors), SPOT 1-2-3-4 satellites, Shuttle XSAR missions 1-2-3 (SRL 1, 2 and SRTM) and EOSAM/EOSPM mission 1 (MODIS sensor).
This document is also providing in Annex A the file naming conventions for the same A.C.S. DLT formats archived on disk. This format is suitable mainly for NRT productions and temporary storage rather than long term archiving, but the complete binary matching with the DLT format is guaranteed.

1.2 REFERENCE DOCUMENTS

DR1 : 754 - 1985 IEEE Standard for binary floating point arithmetic's (Reaff 1990) [SH10116]

DR2 : LANDSAT 5 TO GROUND STATION INTERFACE DESCRIPTION - Rev 8 Goddard Space Flight Center - GSFC 435 D - 100 - June 84

DR3 : LANDSAT 7 DATA FORMAT CONTROL BOOK (DFCB) - VOLUME IV Goddard Space Flight Center - Rev H - Feb 98

DR4 : SATELLITE DATA TRANSCRIPTION – TRANSCRIBED DATA FORMAT ON DLT - Advanced Computer Systems (ACS) – Version 5.2 - June 2000

DR5 : JAPANESE ERS-1 TO GROUND STATION INTERFACE DESCRIPTION Rev 2 NASDA Earth Observation Center – HE88023 - October 90

DR6 : ERS-1 Satellite to Ground Segment Interface Specification ER-IS-ESA-GS-0001 Issue 2C - 20 Feb. 91

DR7 : SPOT TO DIRECT RECEIVING STATION INTERFACE DOCUMENT Edition 1 Rev 1 CNES and SPOT IMAGE , Proj. Ref. S-IF-0/E-10-SI , 11/12/1995.

DR8 : SPOT TO DIRECT RECEIVING STATION INTERFACE DOCUMENT Edition 2 Rev 0 CNES and SPOT IMAGE , Proj. Ref. S-IF-0/E-10-SI , 01/12/1997.

DR9 : Programming Interfaces between DRS and SPOT IMAGE Edition 0 Rev 0 SPOT IMAGE , Proj. Ref. S-CI-0/E-10-SI , 20/01/1999.

DR10 : Instrument Operations manual Dornier , MA-2042-1200-DS/01, Issue 1, Date 15.03.93

DR11 : MODIS COMMAND, TELEMETRY, SCIENCE AND ENGINEERING DESCRIPTION, Santa Barbara Research Center, Cage Code 11323, Date 14.05.1997

1.3 DEFINITIONS, ACRONYMS AND ABBREVIATIONS

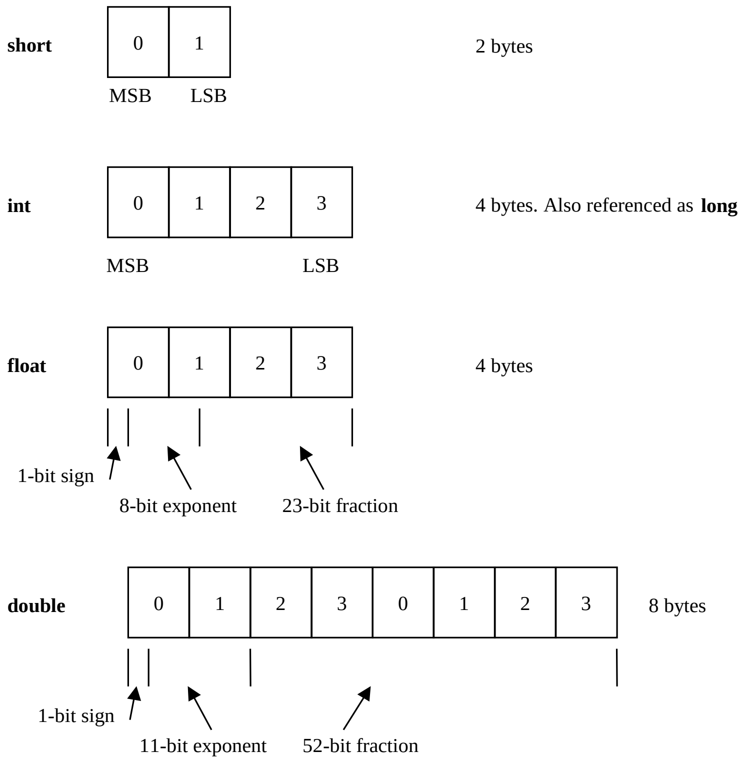
AMI	:	Active Microwave Instrumentation.
CADU	:	Channel Access Data Unit.
CCT	:	Computer Compatible Tape.
DI	:	Direct Ingestion.
DLT	:	Digital Linear Tape.
EOF	:	End of File Tape Mark.
ETM+	:	Enhanced Thematic Mapper Plus.
FD	:	Frames of Data
HDDR	:	High Density Data Recorder.
HDDT	:	High Density Data Tape.
IDHT	:	Instrument Data Handling and Transmission System.
LSB	:	Least Significant Bit.
MODIS	:	Moderate Resolution Imaging Spectroradiometer.
MSB	:	Most Significant Bit.
MSS	:	Multi Spectral Scanner.
PCD	:	Payload Correction Data.
QL	:	Quick Look.
SAR	:	Synthetic Aperture Radar
SRTM	:	Shuttle Radar Topographic Mission
SWIR	:	Short Wave Infrared Radiometer.
TM	:	Thematic Mapper.
TBC	:	To Be Confirmed.
TBD	:	To Be Defined.
WILMA	:	Wide Long Term Multi-Satellite Archive

1.4 GENERAL

1.5 DATA TYPES

In order to avoid confusion and ambiguity we report here a table containing the explanation of standard Data Types which are used in this document. For further details refer to DR1.

Type	Length (in bytes)	Explanation
char	1	Used for character strings.
u_char	1	Used for 8-bits telemetry data (unsigned 8-bit variable)
short	2	Integer 16-bit value (signed) u_short is the unsigned version
int	4	Integer 32-bit value (signed) u_int is the unsigned version
long	4	Integer 32-bit value (signed) - fully equivalent to int u_long is the unsigned version
float	4	Floating point value (32 bit)
double	8	Floating point value (64 bit)



1.6 DEFINITIONS

The following definitions reflect the terminology used by ACS for the Multi-Satellites Data Transcription Systems.

File
A file on DLT cassette is defined as the portion of tape included between two File Marks.

Transcription Area
The Transcription Area is the Transcription Unit. It is a portion of tape, structured in files, as described in the present document, recorded in the Storage Area of the DLT cassette, containing a complete transcribed satellite imaging sequence (a passage), with all related information.

Segment
A satellite pass may be divided into multiple segments. Each segment is stored consecutively within the Sensor Acquired Data File. A Segment Descriptor File contains individual description records for each segment of the pass. A segment is made up of multiple Blocks.

Block
A block is the size unit of transcribed data. A block is made up of multiple swaths.

Swath
The swath is unitary portion of satellite downlinked information. Its definition varies with satellite sensor.
As example:

- for LANDSAT TM and ETM+ the swath is the complete sweep of data (Forward or Reverse), also named Major Frame;
- for J-ERS SAR is the amount of data between two Synchronisation Patterns, named a Line;
- for J-ERS VNIR is includes a fixed number (16) of lines of observed and calibration data
- for ERS 1-2 SAR the swath depends on the Format Synch that was used for the Transcription. It includes a fixed number of formats, with a counter of valid formats on top.
- for SPOT 1-4 is the logical record of the pass. It is named a Line and contains Auxiliary data and Video data;
- for XSAR, as the raw line can change length depending on the PRF (SRL 1,2) the swath length each time contains a different number of raw line formats.
- For EOS Modis the swath is one complete mirror turnaround (1.477 seconds scan) during day or three during night.

Format

- For LANDSAT 7 ETM+ it refers to one of the two virtual channels, running concurrently on the same physical link:
  - Format 1 carries bands 1 through 5 and IR band 6 (low gain).
  - Format 2 carries band 7, IR band 6 (high gain) and Panchromatic.
- In ERS terminology a format consists of 29 frames of 256 bytes each, which includes raw downlink range lines and ancillary data.
- For SPOT 1-2-3-4 a format (or frame) is the response of the array of sensors constituting HRV (or HRVIR) instruments. A format represents one channel, de-scrambled and DPCM decoded, video data, excluded auxiliary data.
- In XSAR it is a single raw line as recorded by the sensor.

Reserved
The portion described as Reserved contains information that is not necessary for the reading of the cassette and the comprehension of its contents. This information is only usable by ACS modules, for further data processing. Any non-ACS software producing the format described in this document should fill these fields with null characters (0 Hex).

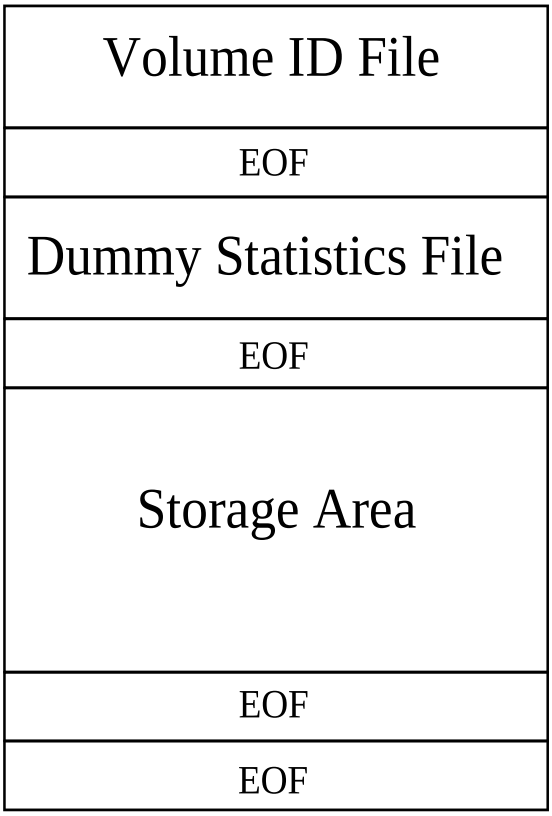
Filler
The portions named as Filler are empty portions, mainly used to pad the record to the specified length.

2. TRANSCRIBED DATA

The DLT cassette tape has the physical structure described below.
It is divided in two logical zones:

- The first one is a Volume ID File which is used to identify the tape and to describe it;
- The second one is the Storage Area containing the transcribed data. It contains "N" Transcription Areas.

This structure is completely independent from the specific kind of data (satellite data in the present case) being recorded. The two sections are readable and writable by the user, with user-defined contents described in the next paragraphs. Hidden portions of the tape contain reserved information used by the recorder procedure. The two zones are divided by a Dummy Statistics File (see paragraph for the Statistics File definitions).
The following figure depicts the global physical structure of the DLT cassette tape.



2.1 VOLUME ID. FILE

This File is dedicated to the identification of the DLT cassette.
The Volume ID contains a set of data organised according to the following structure:

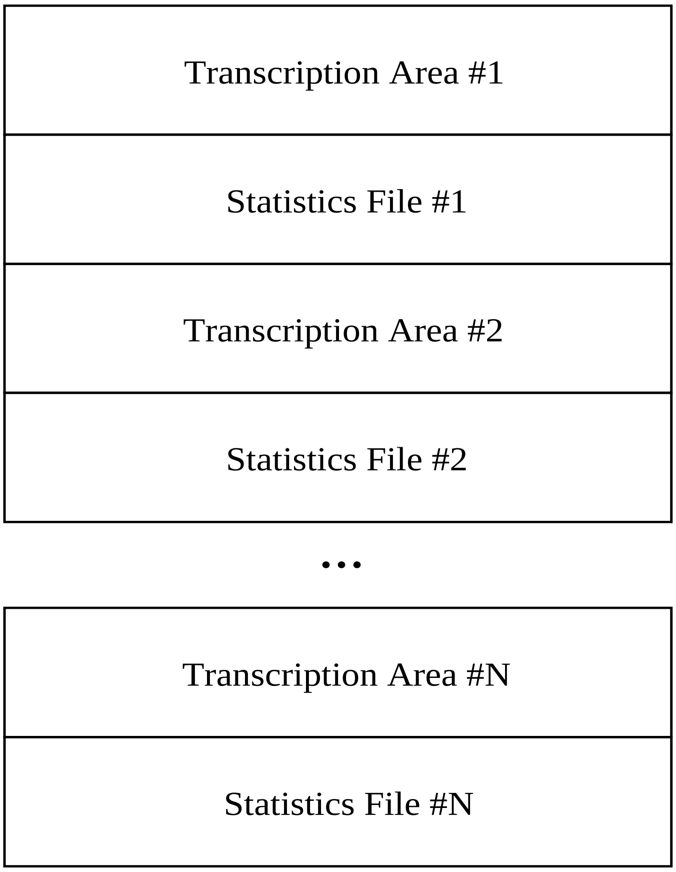
Field	Start	Stop	Bytes	Type	Description
Tape Key	1	4	4	u_int	Tape Key defined differently per satellite : 0x29111996 = LANDSAT MSS 0x30111996 = LANDSAT TM 0x30111996 = LANDSAT ETM+ 0x30071998 = J-ERS 0x28111996 = ERS SAR 0x30071996 = SPOT HRV(IR) 0x30012000 = SHUTTLE(XSAR) 0x23062001 = EOS The operator, while initialising the new tape, defines the key according to the satellite and sensor that he is going to transcribe over.
Tape_Number	5	8	4	u_int	Number assigned to the present cassette
Tape_in_Date	9	40	32	char	Tape Initialisation Date in the syntax : ThuNov16$18:29:34$1995$$$$$$$$ where $ represents the blank character.
Reserved	41	112	72	char	Reserved
Reserved	113	124	12	3*u_int	Reserved
Filler	125	160	36	char	Filler

2.2 DUMMY STATISTICS FILE

This File is fully equivalent to a Statistics File (described in 3.5) but with only one empty record. Its presence permits the generalization of tape inspection procedure (described in 3.5) also when DLT cassettes without any transcription are mounted.

2.3 STORAGE AREA

The Storage Area is logically subdivided into Transcription Areas, according to the following structure:



Each Transcription Area contains the data of one complete passage, as acquired by the station, and many parameters related to satellite orbit, to pass acquisition and to storage structure.

The Transcription Area contains **at least five** files:

- the first file named “Pass Identification Header”;
- the second file named “Sensor Acquired Data”;
- the third file named “User Header”;
- the fourth file named “Segment Descriptor”;
- an end file named “Block Address Descriptor”.

These files are mandatory for the WILMA format described in this document.

Other auxiliary files may be included between the "Segment Descriptor" and the "Block Address Descriptor", depending on specific satellite transcription:

- an "Orbit Data" file;
- a "Payload Correction" file.
- a "Raw Header" file
- an "Attitude" file
- a "Processing Parameter" file

The various files are separate by EOF, hereafter indicated by "Mark".

At the end of each Transcription Area a new file is inserted, named "Statistic File". The file corresponding to Transcription Area ***k*** has (*k+1)* records. It has a Dummy Statistic record of content equivalent to the Dummy Statistics file saw before, and ***k*** Statistic records with information's on all the Transcriptions present on tape. The Statistic File in this format can be seen like a tape directory.
The output data description in the following pages is, unless explicitly stated, valid for all of the satellite whose data are transcribed in ACS systems. Specific features are clearly outlined.

The logical structure of this format foresees the "Sensor Acquired Data" file (containing optical or SAR data and all auxiliary included in the acquired data) at the beginning of the Transcription Area, only preceded by a limited amount of information (Pass Identification Header). The position of Sensor Data at the top of tape portion allows the real time writing during the transcription, leaving auxiliary information extraction and formatting for a post processing step.

2.3.1 DATA RETRIEVAL METHOD

The Transcription Area is designed to allow the fast pointing to Sensor Data. The mechanism used to retrieve a given standard frame inside a file is the following.
Each Transcription Area corresponds to one satellite pass. The "Sensor Acquired Data" file of the Transcription Area is subdivided in Blocks. Each Block is described (in the "Block Address Descriptor") by one Block Address Unit containing the satellite time of that block.
Besides, each frame of a pass is characterised by its start time.
The frame retrieval is thus accomplished in three steps:

1. Searching in the "Block Address Descriptor" for the block with time closest (and preceding) the (known) desired frame start time.
2. Positioning at the address found at step 1.
3. Forward scanning of the swaths, reading the swath-time (contained in the auxiliary information) in order to reach the start time of the desired frame.

IMPORTANT REMARK :

It must be pointed out that in this document we use the term "record" to mean "block of data inside a file"
It must be also pointed out that in some points of the following description, the number of bytes effectively used for a given type of data exceeds the theoretical one. This is due to the padding rules used in the standard C structures.

2.4 TRANSCRIPTION AREA

2.4.1 INTRODUCTION

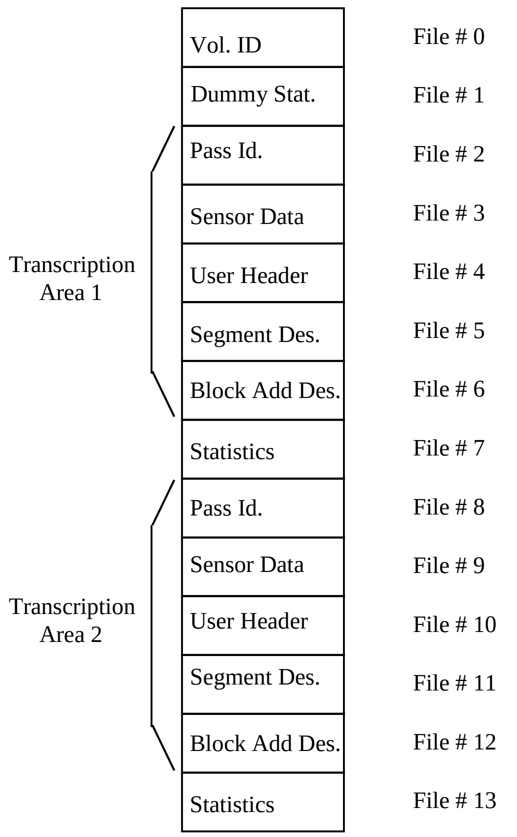
The general principle of the data transcriptions on DLT is that the Sensor Data file is always preceded by a “Pass Identification Header” which contains all the information available at the beginning of the pass: it is thus only partially filled. This file is written immediately before the acquisition of satellite data and is useful when reading the tape in order to know which kind of data follows.
The “User Header” file follows the Sensor Data file. It has the same structure as the “Pass Identification Header”, but is filled in all its fields.
The method used to address information is to count the EOF marks skip. Each mark refers to a file, and the numbering is absolute and zero relative in the DLT. Skipping K marks from the beginning of the tape turns in pointing to file K + 1.
A “Segment Descriptor” file always follows the “User Header” file.
The Transcription Area ends with a “Block Address Descriptor”, needed in the “Data retrieval method “ described in paragraph 3.3.1. These four files have the same structure in all satellites/sensors transcribed.
The other files structure and content depends on the specific satellite, and their difference may be found in following sections.
In the next paragraphs we present briefly the structure of the Transcription Area for each satellite or sensor managed in this document. They should be intended only to give a quick look to what will be better detailed in the following paragraphs.

2.4.1.1 LANDSAT, EOSAM, J-ERS (VNIR)

The standard (minimum and mandatory) files succession in DLT Cassette is described by the following figure. It is applicable for:

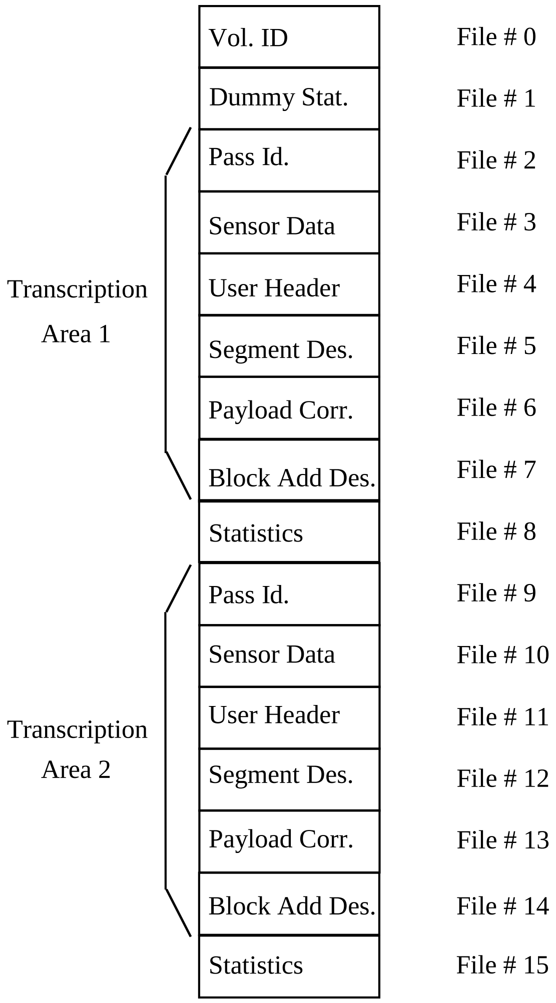
- LANDSAT (TM, MSS, ETM+);
- EOSAM (MODIS)
- J-ERS (VNIR)

No other auxiliary files are added to the common four files "Pass Identification Header", "User Header", "Segment Descriptor", "Block Address Descriptor" and to the "Sensor Acquired Data".



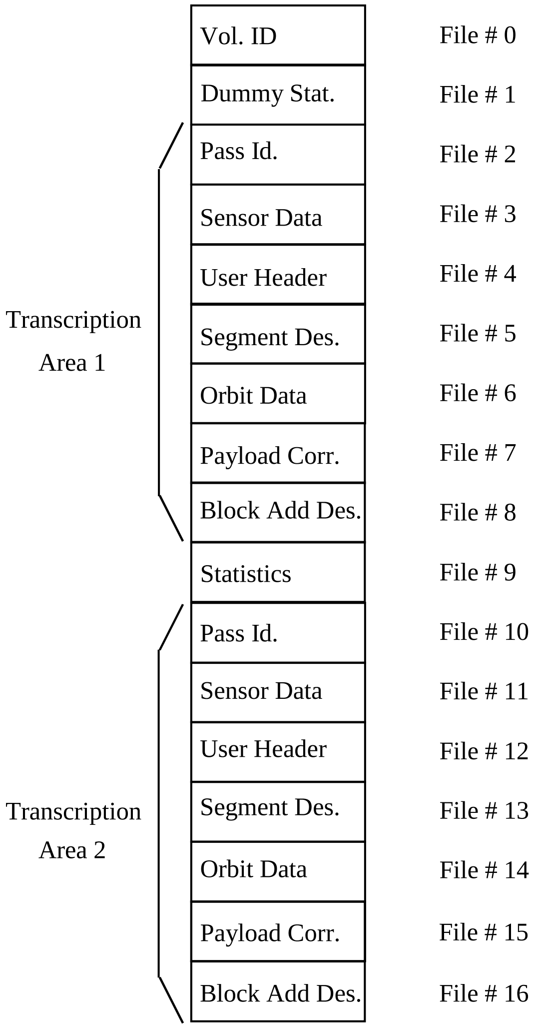
2.4.1.2 EOSPM (AQUA)

The files succession in DLT Cassette described by the following figure is applicable only for EOSPM (MODIS sensor). Six of the seven files enumerated in section 3.3 are present: Orbit Data file is absent.



2.4.1.3 J-ERS (SAR)

The files succession in DLT Cassette is described for J-ERS (SAR Sensor) by the following figure.
All the seven files enumerated in section 3.3 are present.

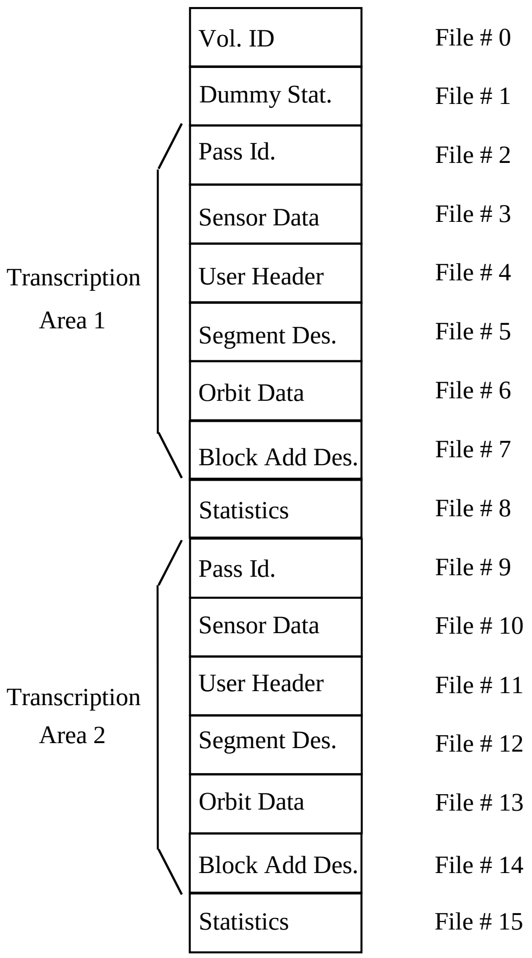


2.4.1.4 ERS SAR, SPOT, SHUTTLE XSAR

The files succession in DLT Cassette is described for:

- ERS 1/2 (SAR Sensor);
- SPOT 1-2-3 (HRV Sensor) and for SPOT 4 (HRVIR Sensor);
- XSAR 1-2-3 (SRL 1,2 and SRTM);

by the following figure.
Six of the seven files enumerated in section 3.3 are present: Payload Correction file is absent.



2.4.2 “PASS IDENTIFICATION HEADER” FILE

The “Pass Identification Header” is the first file of the Transcription Area. It contains the information, available at the beginning of the transcription, to unambiguously identify the imaging sequence contained in the file.
This record is divided into five **logical sections**:

1. Mission and Instrument Identification
2. Ground Stations and Transcription System Identification
3. Transcription Identification
4. Orbit and Acquisition Identification
5. Pointers to Tape Data Structure

In the following pages the logical sections of the “Pass Identification Header” are described in detail.
It must be pointed out that some fields are applicable only in case of transcription from HDDT.

Mission and Instrument Identification

Field	Start	Stop	Bytes	Type	Description
Reserved	1	76	76	char	Reserved
Satellite_ID	77	78	2	u_short	Satellite Code ([1])
Mission_ID	79	80	2	u_short	Satellite Mission Number
Instr_Type_ID	81	82	2	u_short	Instrument Type Code ([2])
Reserved	83	86	4	2*u_short	Reserved

Ground Stations and Transcription System Identification

Field	Start	Stop	Bytes	Type	Description
Station_ID	87	88	2	short	Acquisition Ground Station Code ([3])
Station_DT_ID	89	90	2	short	Data Transcription Ground Station Code
Filler	91	92	2	char	Padding for stucture alignment

Transcription Identification

Field	Start	Stop	Bytes	Type	Description
Source_Type	93	96	4	u_int	Data Source Type ([4])
Format_SyncType	97	100	4	u_int	Format Synchroniser/Decommutator Code ([5])
Reserved	101	108	8	2*u_int	Reserved

Orbit and Acquisition Identification

Field	Start	Stop	Bytes	Type	Description
Track(DataTake)_Number	109	112	4	int	Track or Data Take Number (when applicable)
Orbit_Number	113	116	4	int	Orbit number (when applicable)
Reserved	117	152	36	9*int	Reserved
Reserved	153	158	6	3*short	Reserved

[1] Codes for **Satellites** are provided in. 3.1

[2] Codes for **Instrument Types** are provided in 3.2

[3] Codes for **Station Names** are provided in 3.5

[4] Codes for **Source Type** are provided in 3.3

[5] Codes for **Format Synchroniser** are provided in 3.4

Reserved	159	176	18	char	Reserved
Transcription _Date	177	182	6	3*short	Transcription Date in D M Y (WARNING: Year could be expressed in some tapes as years from 1900)
Reserved	183	194	12	6*short	Reserved
Filler	195	196	2	char	Padding for stucture alignment

Pointers to Tape Data Structure

Field	**Start**	**Stop**	**Bytes**	**Type**	**Description**
Reserved	197	216	20	5*int	Reserved
Physical_Address_1	217	220	4	long	User Header Address[6]
Physical_Address_2	221	224	4	long	Pass Id. Header Address[7]
Reserved	225	876	652	char	Reserved

[6] File Number of the User Header. It is the number of files, numbered from DLT start, preceding the User Header file in current Transcription Area.

[7] File Number of the Pass Id. Header. It is the number of files, numbered from DLT start, preceding the Pass Id. Header file in current Transcription Area.

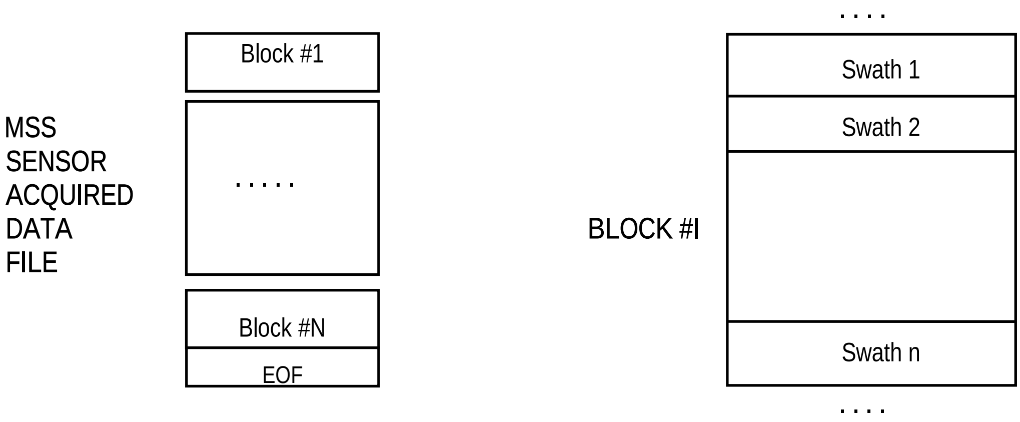
2.4.3 "SENSOR ACQUIRED DATA" FILE

The Video Data are transcribed into tape as they are received in the memory of the host computer. Their format depends on the specific type of sensor.
There is some dependence in the structure of transcribed data block on the Format Synchroniser that has been used for the ingestion of the data into the computer memory. In fact, Format Synchronisers do not manipulate the data, but some reformatting is sometimes executed. This implies that two transcription files of the same pass, made by Transcription systems that use different Format Synchronisers, may slightly differ in marginal aspects. This is the reason why we put, in the fields of the Header files of the Transcription Area, the code of the Format Synchroniser used for the transcription.
The reproduction software must take care of this information to manage the transcribed data. Following paragraphs detail the various cases.
Anyhow, up to now all ACS facilities writing on DLT are using the ACS software Frame Synchroniser installed in the Direct Ingestion systems and they have all the same data format, according only on the satellite and the sensor.

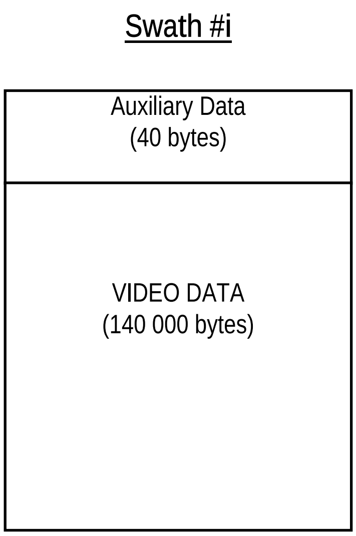
2.4.3.1 LANDSAT MSS “SENSOR ACQUIRED DATA”

As mentioned before the structure of the transcribed data depend on the Format Synchroniser used for data ingestion into the host computer before transcription. In the case of LANDSAT MSS data the system uses the format adopted by the ACS Software Format Synchronizer.

The LANDSAT MSS Sensor Data are written on tape as follows. The data corresponding to one pass are subdivided in Blocks having the same length (about 4 Megabytes) to allow the pointing mechanism described in previous sections. Each block contains a fixed number of entire swaths. This number has to be read from the “User Header”.



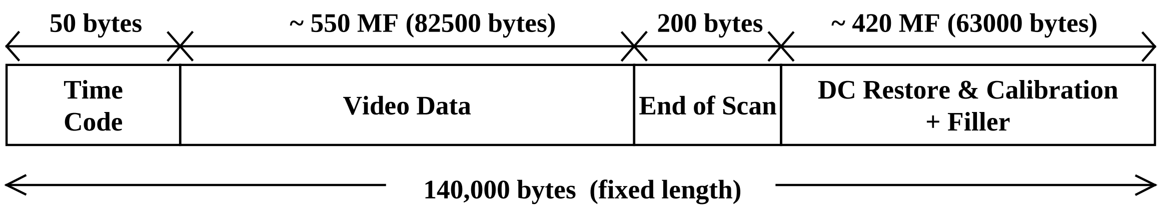
Each swath is 140,040 bytes long and consists of the following portions of data:



The Auxiliary Data structure reports information extracted from swaths data during the transcription procedure. It appears as follows :

Field	Start	Stop	Bytes	Type	Description
Day	1	4	4	int	# of days from the current year
Hours	5	8	4	int	# of hours from midnight
Minutes	9	12	4	int	# of minutes
Seconds	13	16	4	int	# of seconds
Milliseconds	17	20	4	int	# of milliseconds
msec fractionary	21	22	2	u_short	# of 1/16 of milliseconds
Satellite Id.	23	24	2	u_short	LANDSAT mission number
Line Length	25	28	4	int	Length of active scan in Minor Frames unity
Swath Length	29	32	4	int	Length in bytes of a swath (as transmitted from satellite. Note that the transcribed swath is a fixed part of this length).
Satellite Time	33	40	8	double	Swath time in milliseconds and 1/16 of milliseconds from beginning of year .

The Video Data structure on DLT is 140,000 bytes long (fixed) and has the following form:

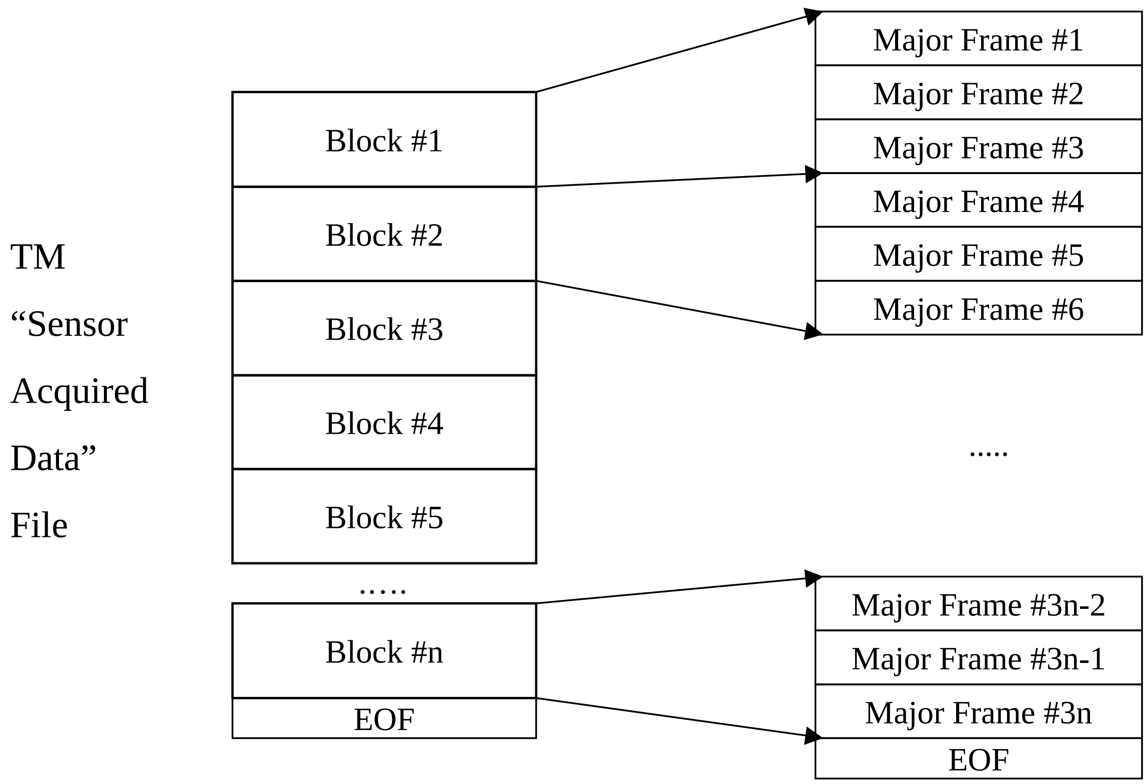


One Minor Frame is 150 bytes long and is an exact copy of the original from satellite.
For more details refer to DR2 document.

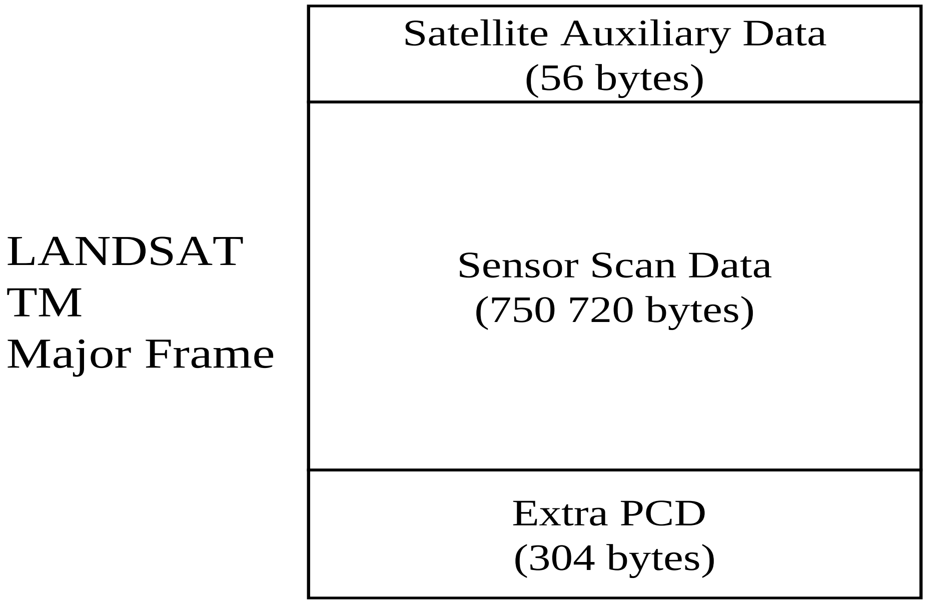
2.4.3.2 LANDSAT TM "SENSOR ACQUIRED DATA"

As mentioned before the structure of the transcribed data depend on the Format Synchroniser used for the data ingestion into the host computer before transcription. In the case of LANDSAT TM data the system uses the format adopted by the ACS Software Format Synchroniser.

The LANDSAT TM Sensor Acquired Data are written on tape as follows. The data corresponding to one pass are subdivided in Blocks having the same length (about 2 Megabytes) to allow the pointing mechanism described in section 3.3.1.
The real number of Major Frames per block must be read from the "User Header".
As an example for LANDSAT TM each block contains usually 3 complete Major Frames, as reported below:



Each Major Frame has the length of 751 080 bytes organised as follows:



Detailed description of TM Major Frame structures follows.

TM MAJOR FRAME FIELDS :

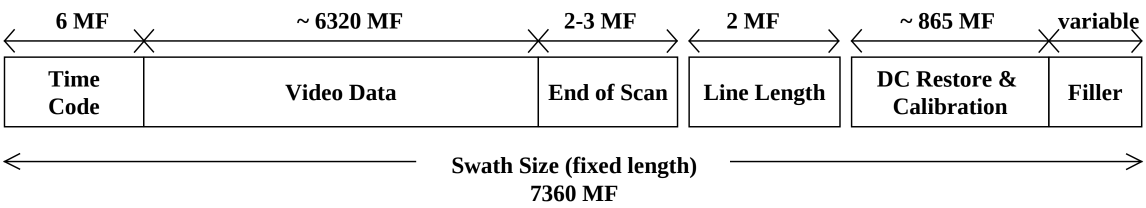
Field	Start	Stop	Bytes	Description
Satellite Auxiliary Data	1	56	56	Satellite Data Extracted from swath during transcription. Useful for swath processing.
Sensor Scan Data	57	750 776	750 720	Scan Data.
Extra PCD	750 777	751 080	304	1 PCD byte extracted from each MF of abolished useless data.

The Satellite Auxiliary Data portion reports information extracted from video data during the transcription procedure. It appears as follows.

Field	Start	Stop	Bytes	Type	Description
Day	1	4	4	u_int	# of days from the current year
Hours	5	8	4	u_int	# of hours from midnight
Minutes	9	12	4	u_int	# of minutes
Seconds	13	16	4	u_int	# of seconds
Milliseconds	17	20	4	u_int	# of milliseconds
msec fractionary	21	22	2	u_short	# of 1/16 of milliseconds
Satellite Id.	23	24	2	u_short	LANDSAT mission number
Fhserr	25	28	4	u_int	First half scan error

Shserr	29	32	4	u_int	Second half scan error
Scan direction	33	36	4	u_int	Direction of Scan: 0x00000000 Forward 0xFFFFFFFF Reverse
Line Length	37	40	4	u_int	Length of active scan in Minor Frames unity
Swath Length	41	44	4	u_int	Length in bytes of a swath (as transmitted from satellite. Note that the transcribed swath is a fixed part of this length).
Filler	45	48	4	u_int	Padding for structure alignment
Satellite Time	49	56	8	Double	Swath time in milliseconds and 1/16 of milliseconds from beginning of year .

The Sensor Scan Data portion (also called swath) reports information as downlinked from satellite, with only few reformatting in order to suppress useless swath portions (Scan Line Start and filler portion). The data unit of a swath is the Minor Frame (hereafter indicated by MF) of 102 bytes, including 4 bytes of sync, 1 byte of band 6 video data,1 of PCD and 96 of sensors video data for the other bands. This structure is an exact copy of the original from satellite.
The detailed structure of a Swath follows:

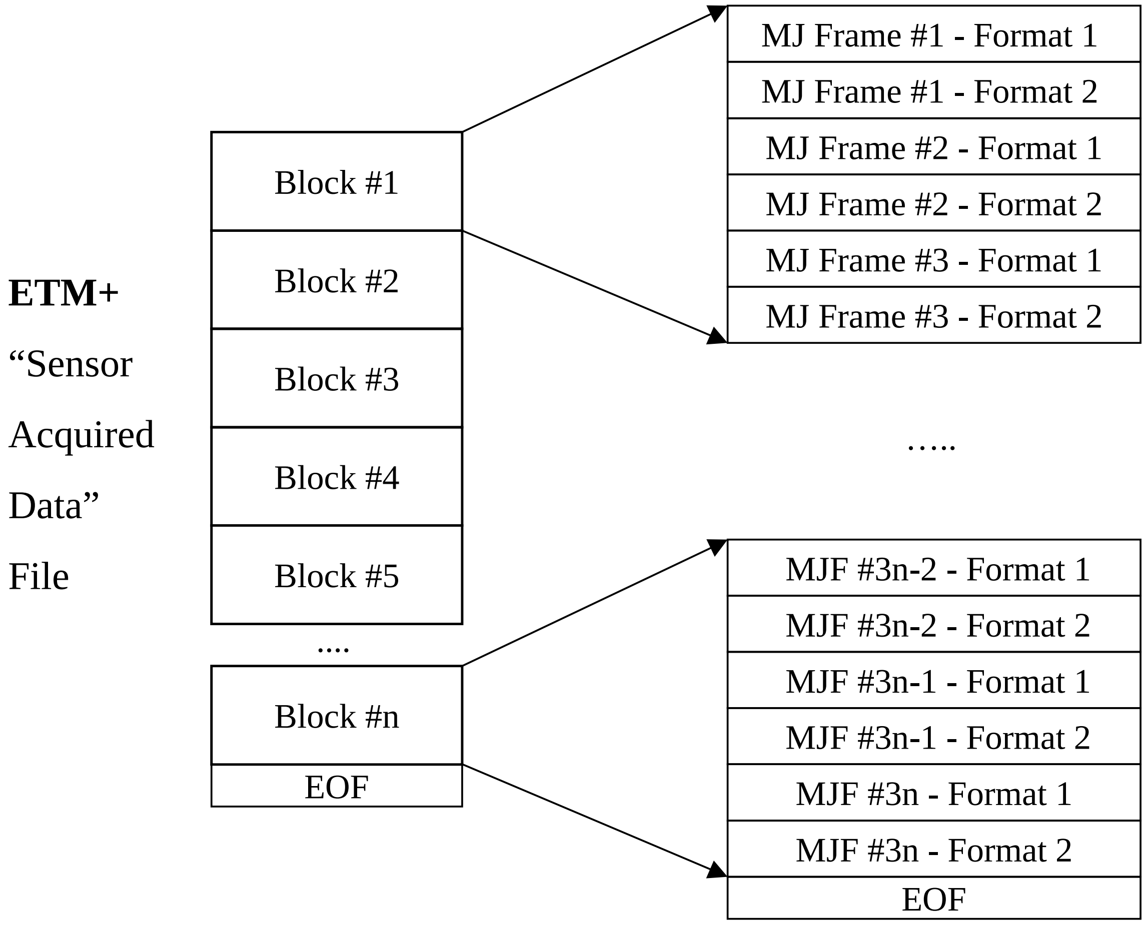


The Extra PCD data are 304 PCD bytes extracted from the abolished swath portions (filler portion).

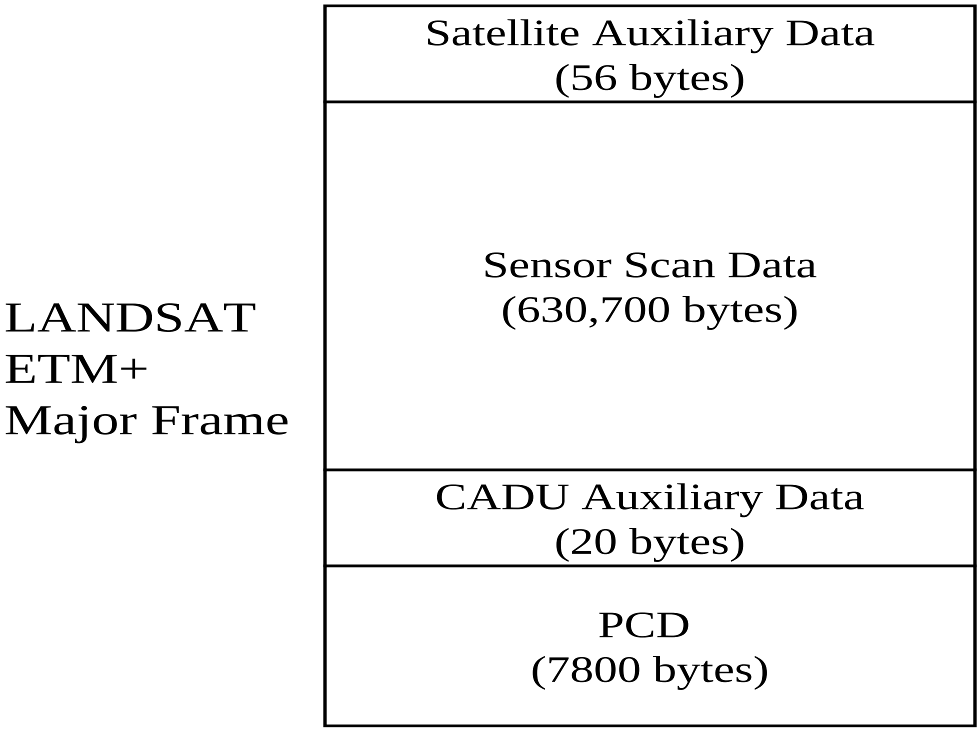
2.4.3.3 LANDSAT 7 ETM+ “SENSOR ACQUIRED DATA”

In the case of LANDSAT 7 ETM+, as for LANDSAT TM, data are transcribed using the format adopted by the ACS Software Format Synchroniser.

The LANDSAT 7 ETM+ Sensor Data are written on tape as follows. The data corresponding to one pass are subdivided in Blocks having the same length (about 4 Megabytes) to allow the pointing mechanism described in section 3.2.1.
Each Block is formed by a fixed number of Major Frames (MJF), format interleaved. This number has to be read from the “User Header” file. Here is reported an example with the most common value of 6 MJF per block.



Each MJF contains a complete sweep (Forward or Reverse) of ETM+ instrument (swath) and some other auxiliary information. It has the length of 638,576 bytes, organised as follows:



Detailed description of MJF structures follows.

ETM+ MAJOR FRAME :

Field	Start	Stop	Bytes	Description
Satellite Auxiliary Data	1	56	56	Satellite Data Extracted from swath during transcription. Useful for swath processing.
Sensor Scan Data	57	630,756	630,700	Scan Data (called swath).
CADU Auxiliary Data	630,757	630,776	20	CADU auxiliary information.
PCD	630,777	638,576	7800	10 PCD bytes extracted from each CADU.

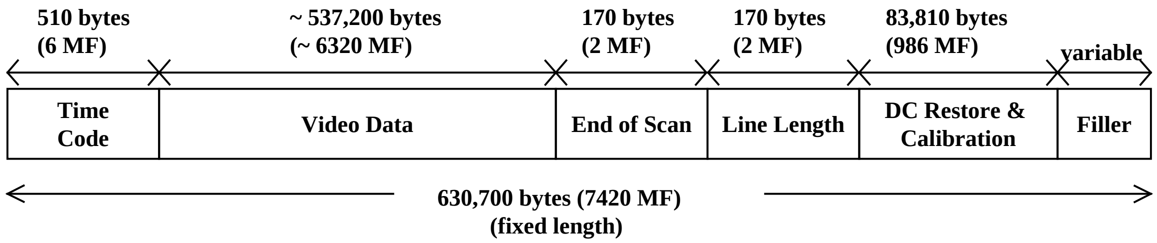
The Satellite Auxiliary Data structure reports information extracted from video data during the transcription procedure. It appears as follows.

Field	Start	Stop	Bytes	Type	Description
Day	1	4	4	int	# of days from the current year
Hours	5	8	4	int	# of hours from midnight
Minutes	9	12	4	int	# of minutes
Seconds	13	16	4	int	# of seconds
Milliseconds	17	20	4	int	# of milliseconds
msec fractionary	21	22	2	u_short	# of 1/16 of milliseconds
Satellite Id.	23	24	2	u_short	LANDSAT mission number

Format Id.	25	25	1	u_char	Format Identifier: 0x00 FMT 1 0xFF FMT 2
Spare	26	28	3	3*u_char	Available fields
Fhserr	29	32	4	int	First half scan error
Shserr	33	36	4	int	Second half scan error
Scan direction	37	40	4	int	Direction of Scan: 0x00000000 Forward 0xFFFFFFFF Reverse
Line Length	41	44	4	int	Length of active scan in Minor Frames unity
Swath Length	45	48	4	int	Length in bytes of a swath (as transmitted from satellite. Note that the transcribed swath is a fixed part of this length).
Satellite Time	49	56	8	double	Swath time in milliseconds and 1/16 of milliseconds from beginning of year .

The Sensor Scan Data portion reports information in swaths form, extracted and reconstructed from the "Mission Data Zone" of CADUs, with only few reformatting in order to suppress useless swath portions (Scan Line Start code and filler portion); the swath obtained in this way is formally equivalent to the TM one (not in size) .
The data unit of a swath is the Minor Frame (hereafter indicated by MF) of 85 bytes, including 4 bytes of IR band 6, 1 spare and 80 of sensors video data. This structure is an exact copy of the original from satellite.
Format 1 or 2 differs in the 80 video bytes content and in band 6 gain. Format 1 represents bands from 1 to 5 with 16 sensors each, and IR band 6 with low gain. Format 2 represents band 7 (16 sensors), Panchromatic (32 sensors sampled twice) and IR band 6 with high gain.

Format 1 and 2 have the same swath structure, here represented:



CADU Auxiliary Data are some useful information about CADUs from which swaths are extracted.

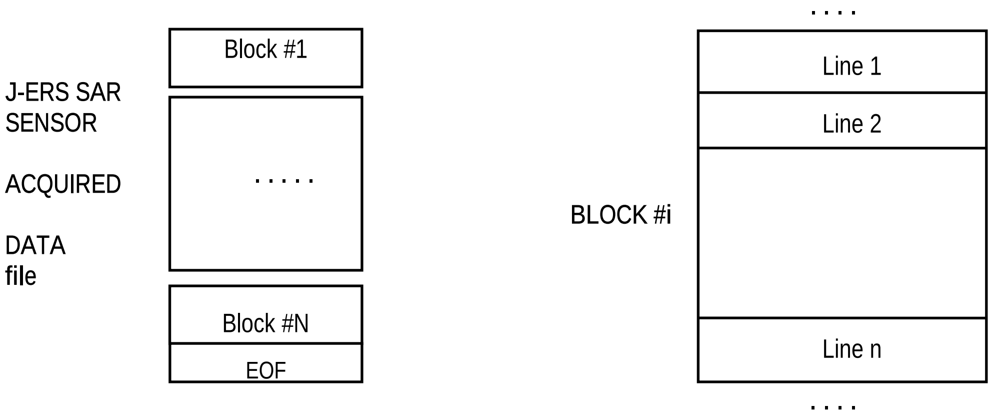
Field	Start	Stop	Bytes	Type	Description
CADU Start Num.	1	4	4	u_int	CADU counter value when SLS occurs.
CADU Stop Num.	5	8	4	u_int	CADU counter value when fill zone ends (at next SLS).
CADU offset	9	12	4	u_int	Bytes offset of SLS from start of Data Zone
Data Priority	13	13	1	u_char	Data priority flag: 0x00 Routine data 0xff Priority data
Spare	14	20	7	u_char	Available field

PCD data are extracted in 10 bytes block from each CADUs and transcribed sequentially. Valid PCD bytes are also extracted from CADU when the abolished swath portion (filler portion) is transmitted, and are contiguous with the others. The PCD field size in MJF is overestimated respect to the needed; the last portion of it will be filled with zeros. In order to recover the right number of valid PCD, the two CADU counter values in previous structure have to be used. The first one represent the CADU counter value when Scan Line Start is detected, the second is the counter value when next SLS is detected. The valid PCD bytes are :

(CADU Stop Num. – CADU Start Num.) * 10

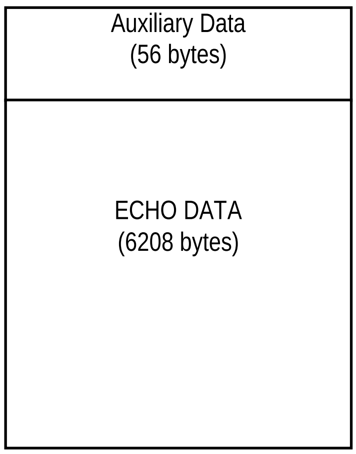
2.4.3.4 J-ERS SAR "SENSOR ACQUIRED DATA"

As for other satellites, the J-ERS SAR Sensor Data are written on tape subdivided in Blocks having the same length (about 2 Megabytes) to allow the pointing mechanism. Each block contains a fixed number of Lines. This number has to be read from the "User Header - Pointers to Tape Data Structure" file.



The J-ERS SAR Observation Data are transmitted as a series of frames. Each frame corresponds to one line of observed or calibration data. Two transmitted channels, named I and Q, have the same Line format and identical auxiliary data content: they differ only for the nature of observed data. In the transcribed data I and Q lines are interleaved each three bits words.
Some auxiliary information and line time are extracted and reported in a Line Header for faster processing procedure. The resulting Line is 6264 bytes long and has the following overall structure:

<u>Line #i</u>



The **Auxiliary Data** structure reports information extracted from downlinked line during the transcription procedure. It appears as follows.

Field	Start	Stop	Bytes	Type	Description
Day	1	4	4	u_int	# of days from 1st January of current year
Hours	5	8	4	u_int	# of hours from midnight
Minutes	9	12	4	u_int	# of minutes
Seconds	13	16	4	u_int	# of seconds
Milliseconds	17	20	4	u_int	# of milliseconds
PRF On/Off	21	21	1	u_char	Pulse Repetition Freq. On = 1, Off = 0
PRF Code	22	22	1	u_char	Code for PRF value: 0x00 = 1505.8 Hz 0x01 = 1530.1 Hz 0x02 = 1555.2 Hz 0x03 = 1581.1 Hz 0x04 = 1606.0 Hz
Calib. Mode	23	23	1	u_char	Calibration Mode On/Off: 0 = All Off 1 = CAL On
Observ. Mode	24	24	1	u_char	Observation mode On/Off: 1 = OBS On 0 = OBS Off
OBS/CAL Start	25	25	1	u_char	"1" is shown at first CAL and first OBS Line
Reserved	26	28	3	3*char	Reserved
Line Counter	29	32	4	u_int	24 bits Line Counter (masked 0x00FFFFFF)
Reserved	33	36	4	u_int	Reserved
Reserved	37	40	4	u_int	Reserved
PRF value	41	48	8	Double	Measured Pulse Repetition Frequency
Line Scan Time	49	56	8	Double	Line duration in milliseconds (1000/PRF)

The **Echo Data** structure on DLT is **6208 bytes** long (fixed).
Each byte of this structure is composed as follows:

0	I2	I1	I0	0	Q2	Q1	Q0

with:

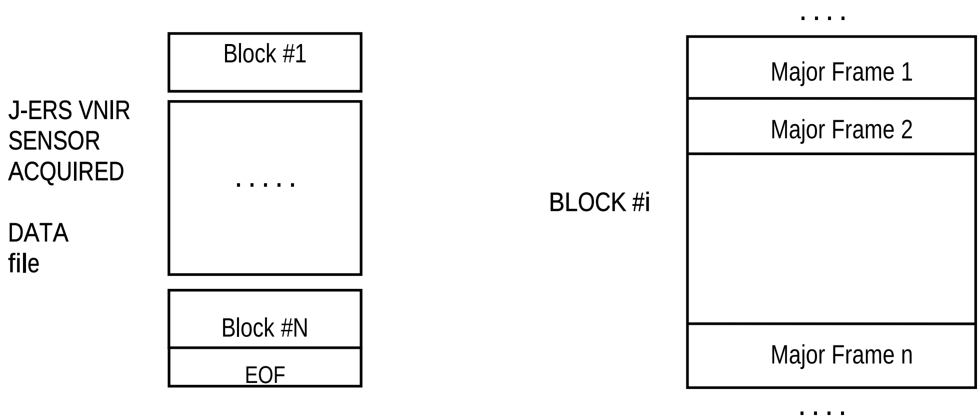
I2 most significant bit and I0 less significant bit of I channel
Q2 most significant bit and Q0 less significant bit of Q channel

Each Echo Data portion on DLT has the following structure. More details on bits meaning can be found in DR5.

From byte	To byte	Bytes	Content	Bits per Channel
0	9	10	Sync code	30
10	32	23	HK data	69
33	40	8	Line Counter	24
41	6197	6157	Data and PCM	18471
6198	6208	10	Random pattern	30

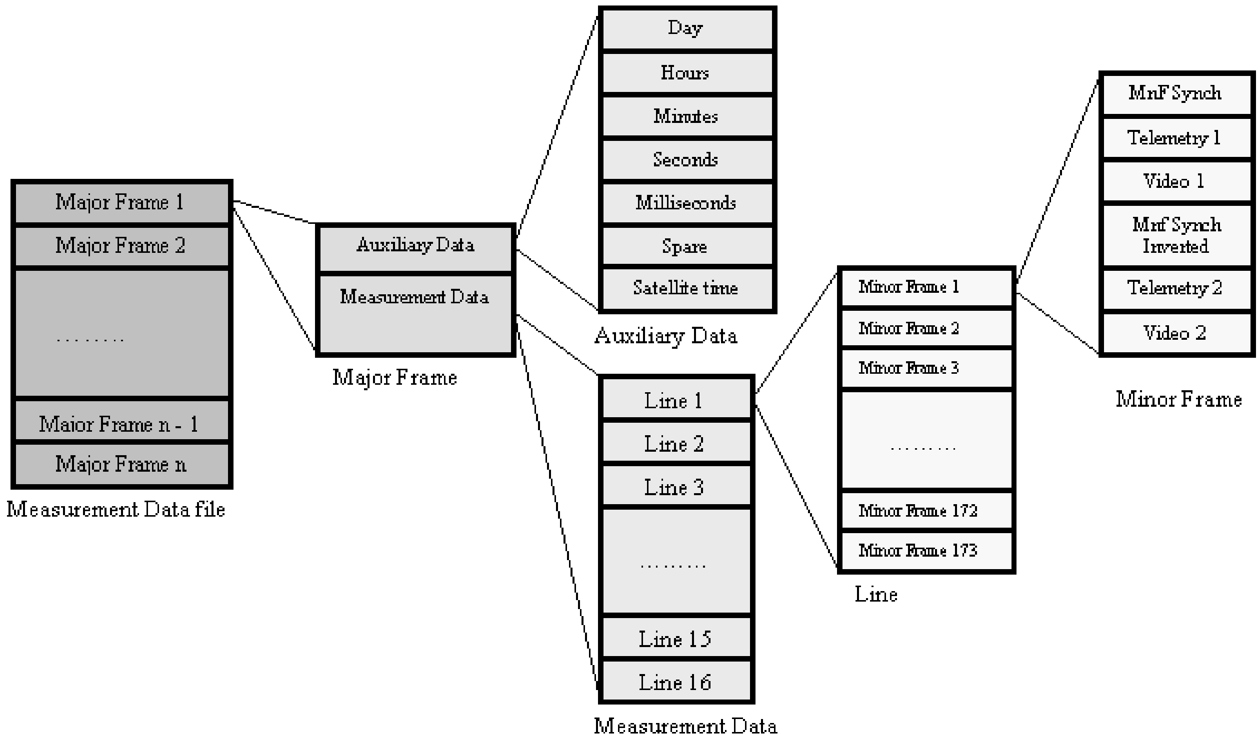
2.4.3.5 J-ERS VNIR “SENSOR ACQUIRED DATA”

The J-ERS VNIR Sensor Data are written on tape subdivided in Blocks having the same length (about 2.2 Megabytes) to allow the pointing mechanism. Each block contains a fixed number of Major Frames (typically 8). This number has to be verified from the “User Header - Pointers to Tape Data Structure” file.



The “Major Frame” is composed by a set of 16 lines of observed and calibration data; each Major Frame (fixed length, 282368 bytes long) contains annotations (32 bytes) and measurement data (282336 bytes).

The following figure provides an overview of the Measurement Data block content:



Field	Start byte	Stop byte	Size (bytes)	Description
Major Frame 1	1	282368	282368	First major frame of observed/calibration data
Major Frame 2	282369	564736	282368	Second major frame of observed/calibration data
……..				
Major Frame n - 1	(n-2)* 282368 + 1	(n-1)* 282368	282368	(n-1)th major frame of observed/calibration data
Major Frame n	(n-1)* 282368 + 1	n * 282368	282368	nth major frame of observed/calibration data

Each Major Frame is 282368 bytes long and subdivided as follows:

Field	Start Byte	Stop byte	Size (bytes)	Description
Auxiliary data	1	32	32	Satellite data extracted from line during transcription
Measurement Data	33	282368	282336	Observed data

The Auxiliary Data field is 32 bytes long and reports information extracted from the corresponding Major Frame during the transcription procedure. Its contents is described in the following table:

Field	Start byte	Stop byte	Size (bytes)	Type	Description
Day	1	4	4	u_int	# of days from 1st January of current year
Hours	5	8	4	u_int	# of hours from midnight
Minutes	9	12	4	u_int	# of minutes
Seconds	13	16	4	u_int	# of seconds
Milliseconds	17	20	4	u_int	# of milliseconds
Spare	21	24	4	u_int	Spare field
Satellite Time	25	32	8	Double	Satellite time (milliseconds from 1st January of current year)

The Measurement Data field is 282336 bytes long and composed by 16 lines of observed /calibration data, as shown in the following table:

Field	Start byte	Stop byte	Size (bytes)	Description
Line 1	1	17646	17646	First line of observed/calibration data
Line 2	17647	35292	17646	Second line of observed/calibration data
…….	……	…….		
Line 15	247045	264690	17646	15th line of observed/calibration data
Line 16	264691	282336	17646	16th line of observed/calibration data

Each **Line**, in turn, is 17646 bytes long and composed by 173 Minor Frames, as shown in the following table:

Field	Start byte	Stop byte	Size (bytes)	Description
Minor Frame 1	1	102	102	
Minor Frame 2	103	204	102	
…….	……	…….		
Minor Frame 172	17443	17544	102	
Minor Frame 173	17545	17646	102	

Each **Minor Frame**, finally, is 102 bytes long and contains the following information (see [DR5] if you need more detailed description):

Field	Start byte	Stop byte	Size (bytes)	Type	Description
Minor frame synch code	1	2	2	u_char	
Telemetry 1	3	3	1	u_char	
Video 1	4	51	48	u_char	
Minor frame synch inverted	52	53	2	u_char	
Telemetry 2	54	54	1	u_char	
Video 2	55	102	48	u_char	

2.4.3.6 ERS SAR "SENSOR ACQUIRED DATA"

The ERS SAR Sensor Data of one imaging sequence are stored consecutively in the file. The characteristics and the quality level of each segment are contained in the "Segment Descriptor" file.
The ERS format, as transmitted from satellite, is described hereafter.

One High Rate Format consists of 29 frames of 256 bytes each. The data of each format are generated by two different sources:

- The IDHT, which generates the synchronization code, the IDHT identification code and the IDHT General Header.
- The AMI, which generates data contained in the auxiliary data field and the AMI Measurement data field.

The structure of an **High Rate Format** is depicted below:

Byte	Content
0 1 2 3	IDHT Sync Code
4 5	IDHT ID
6 -- 15	IDHT General Header
16 --- n	Auxiliary Data
n+1 --- 235	Calibration or Replica Data
236 --- 255	Measurement Data Field

Frame #0

Byte	Content
0 1 2 3	IDHT Sync Code
4 5	IDHT ID
6 --- 255	Measurement Data Field

Frame #1 to #28

For further detail about ERS SAR data format refer to [DR6].

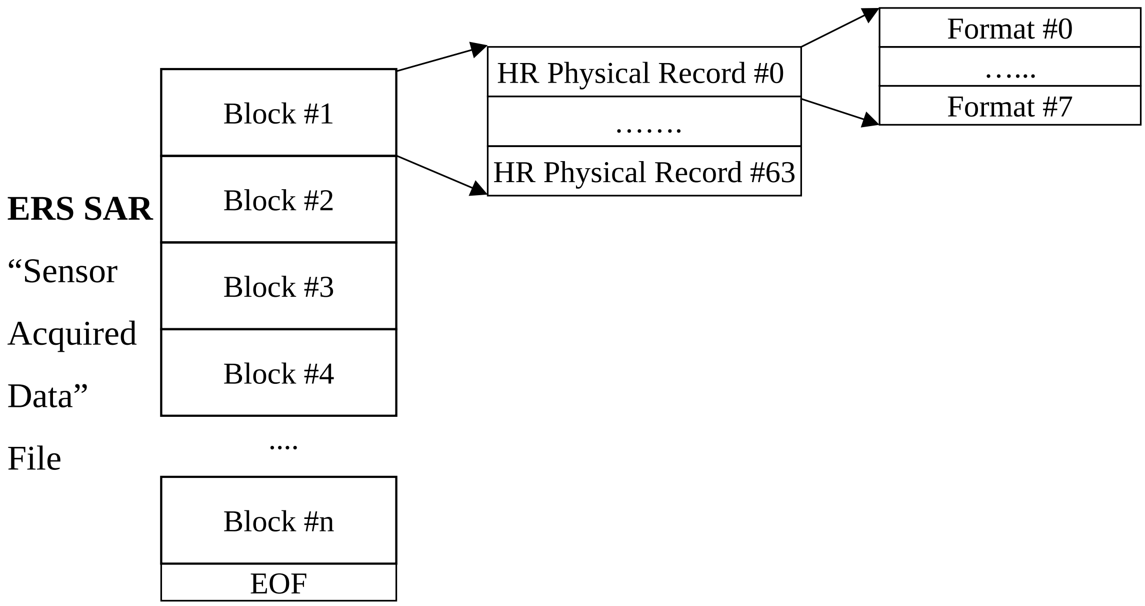
As mentioned before, the format of each satellite raw format depends on the Format Synchronizer used for data ingestion.
In all the DLT transcribed tapes, the ERS SAR ACS Format Synchronizer outputs raw lines exactly in the same format of the ERS High Rate Data structure just mentioned (e.g. 29*256 bytes per format).
ERS Sensor Data are stored in fixed length records, each containing a fixed number of formats.
The fixed length record is the tape physical record of the Sensor Acquired Data file.
The number of formats per record is 8.
The size of the physical record on tape is therefore: (29*256)*8 = 59392 bytes.
Data are transferred without any manipulation. No padding of lost lines is made.

A Block is formed (not physically but logically) by 64 physical records and has the usual length of about 4 Megabytes:

(59392*64 = 3,801,088 bytes).

However, record and block sizes must be read from “User Header file – Pointer to Tape Data Structure”. Specific block characteristics must be read from the “Block Addresses Descriptor” file, as described in section “Data Retrieval Method” 3.3.1.

Formats packing in records and blocks is shown in next figure and summarized in following tables.



Structure of a ERS SAR physical record

Field	Start	Stop	Bytes	Type	Description
Format_0	1	7424	7424	char	First format
Format_1	7425	14848	7424	char	Second format
...			...		
Format_7	51969	59392	7424	char	8th (last) format

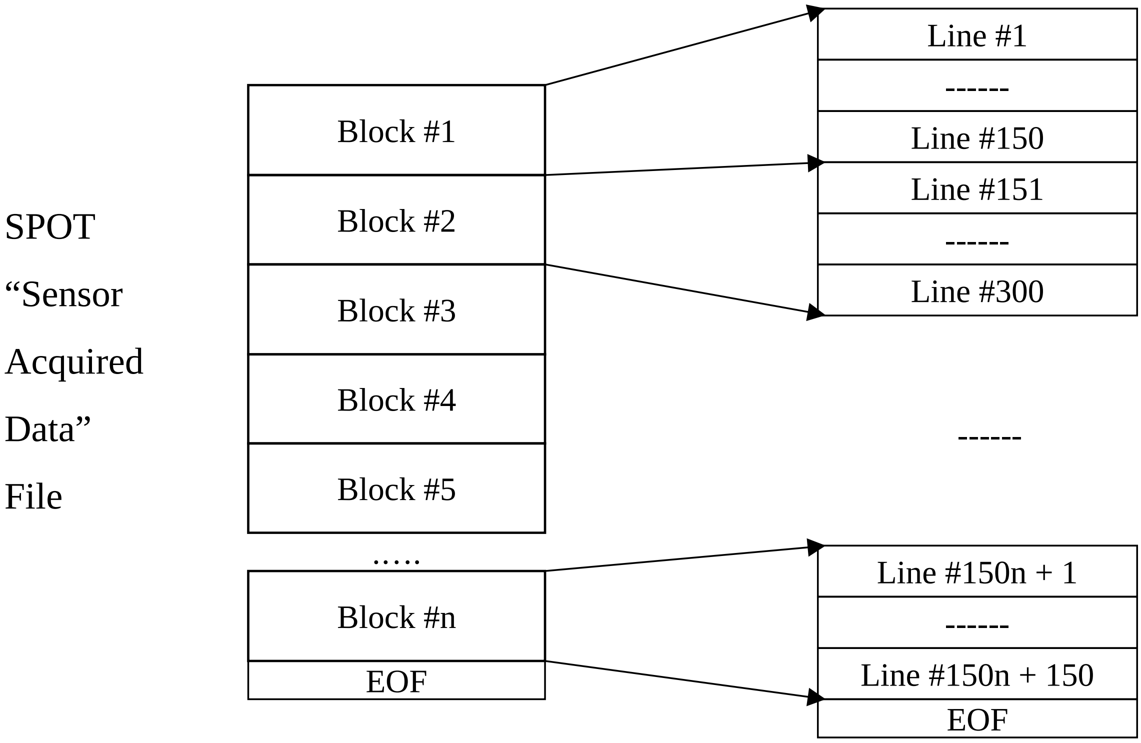
Structure of a ERS SAR (logicaL) Block

Field	Start	Stop	Bytes	Type	Description
Record_0	1	59392	59392	char	First physical record
Record_1	59393	118784	59392	char	Second physical record
...			...		
Record_63	3741697	3801088	59392	char	64th (last) record

2.4.3.7 SPOT "SENSOR ACQUIRED DATA"

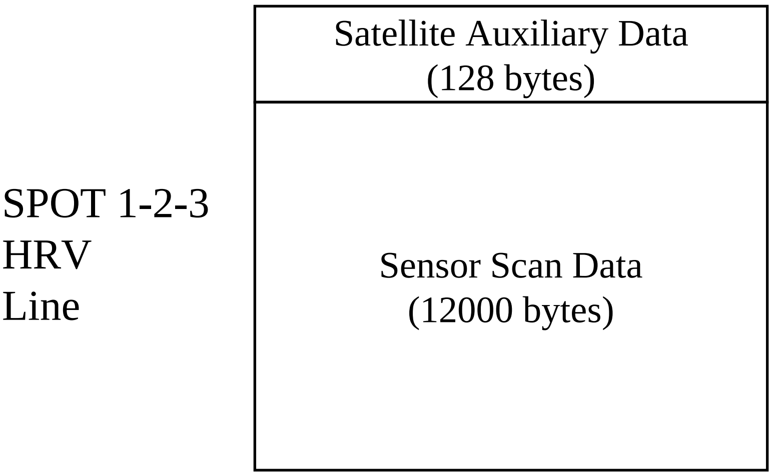
The SPOT 1-2-3 HRV and SPOT 4 HRVIR Sensor Data of one imaging sequence are stored consecutively in the file. Data are separated in two channels and de-scrambled before writing on DLT. The two channels are transcribed on different tapes.
The characteristics and the quality level of each segment are contained in the "Segment Descriptor" file.

SPOT Sensor Data are stored in fixed length records, each containing a fixed number of lines.
The fixed length record is the tape physical record of the Sensor Acquired Data file and represents a Block (as described in the "Block Address Descriptor" file), allowing the pointing mechanism described in section 3.2.1.
Each Block is formed by a fixed number of Lines. The block size and the number of lines per block have to be read from the "User Header" file. Here is reported an example with the most common value of 150 lines per block (and a block size of about 2 Megabytes).



2.4.3.7.1 SPOT 1-2-3 HRV Line

Each SPOT 1-2-3 Line is constituted by a preamble of 128 (8) Bytes plus a data block of 12 000 Bytes as depicted below. Line size is 12128 bytes (8).



The Auxiliary Data structure is the auxiliary data portion preceding each line in downlinked data, with a minimum of changes due to structure alignment. The structure sizes 128 bytes. In the picture below the data are grouped in 16-bit words for the ease of graphical representation (8).

SPOT 1-2-3 Auxiliary Data

1	SYNCHRONISATION WORDS			
2				
3				
4				
5	FMT Id.	FMT Id. (R)	Reserved	Frame # (4 msb)
6	Frame # (16 lsb)			
7	Ch 1 Status		Ch2 Status	
8	Reserved			
9	HRV 1 mode		HRV 2 mode	
10	HRV 1 gains			
11	HRV 2 gains			
12	Reserved			
13	Attitude status		Yaw velocity	
14	Roll velocity		Pitch velocity	
15				
16				
17				
18				
…				

8 Data archived in the South African station of Hartebeeshoek, has the Auxiliary Data structure of 104 bytes, without the reserved fields from 53 up to 64 of the picture shown. The Line size of this archived data is 12104 instead 12128 bytes.

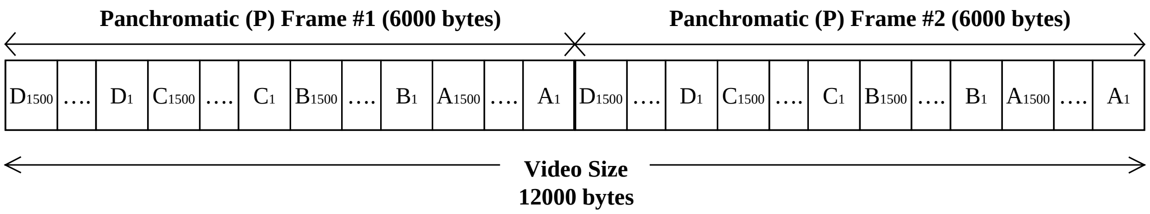
Word		
…	Reserved (34 words)	
…		
45		
46		
47		
48		
49	HRV 1 status	HRV 1 mirror position
50	HRV 2 status	HRV 2 mirror position
51	On board sequence time	
52		
53	Reserved (12 words)	
54		
55		
56		
57		
58		
59		
60		
61		
62		
63		
64		

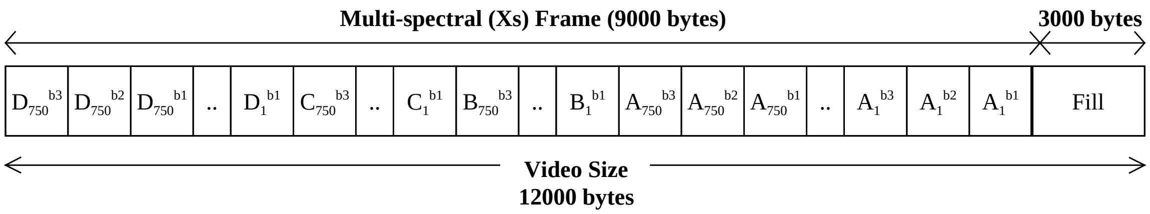
For further details about the fields of the above structure refer to [DR7].

Sensor Scan Data portion following the Auxiliary portion contains Video data, organized in the following way.

- In the panchromatic (P) case there are two successive frames of 6000 Bytes each.
- In the Xs case there is one multi-spectral frame (3 bands x 3000 pixels), organized as band interleaved.

The following picture depicts the concepts described above.

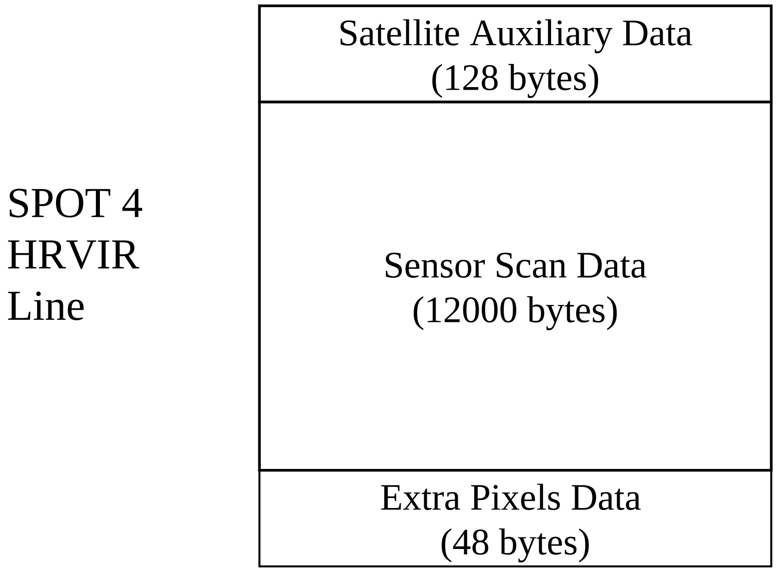




Frames of video data are arranged in order to be West-East oriented (pixel 1=East). To achieve this result the instrument barrettes are inverted.

2.4.3.7.2 SPOT 4 HRVIR Line

Each SPOT 4 (HRVIR sensor) Line is constituted by a preamble of 128 Bytes plus a data block of 12000 Bytes, as for SPOT 1-2-3 Line. At the end of each Line a block of 48 bytes is added, in order to store the bytes representing the extra pixels of each array. These extra pixels are called "recovering detectors" and carry redundant information not used in images, but used for inter-array correction.
SPOT 4 Line is depicted below. Line size is 12176 bytes.



The Auxiliary Data structure is the auxiliary data portion preceding each line in downlinked data, with a minimum of changes due to structure alignment. The structure sizes 128 bytes. In the picture below the data are grouped in 16-bit words for the ease of graphical representation.

SPOT 4 Auxiliary Data

1	SYNCHRONISATION WORDS			
2				
3				
4				
5	FMT Id.	FMT Id. (R)	Reserved	Frame # (4 msb)
6	Frame # (16 lsb)			
7	Ch 1 Status		Ch2 Status	
8	Reserved		HRVIR 1,2 SWIR gain	
9	HRVIR 1 mode		HRVIR 2 mode	
10	HRVIR 1 M, XI gains			
11	HRVIR 2 M, XI gains			
12	Reserved		HRVIR 1,2 working mode	
13	Attitude status		Yaw velocity (8 lsb)	
14	Roll velocity (8 lsb)		Pitch velocity (8 lsb)	
15	reserved			
16	reserved		Velocities (2 msb)	
17	reserved			
18				

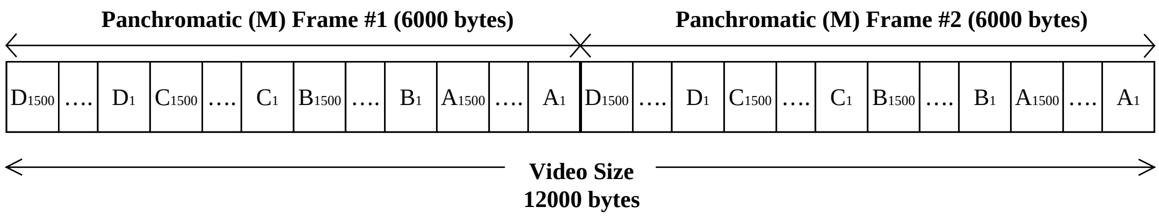
…	Reserved for decoder device
26	
27	Orbital elements
…	
43	
44	DORIS/DIODE Ephemeris
…	
48	
49	HRVIR 1 mirror status / HRVIR 1 mirror position
50	HRVIR 2 mirror status / HRVIR 2 mirror position
51	On board sequence time
52	
53	Reserved for decoder device
54	
55	
56	
57	Reserved (2 words)
58	
59	Time Tagging coefficients
60	
61	
62	
63	
64	Reserved

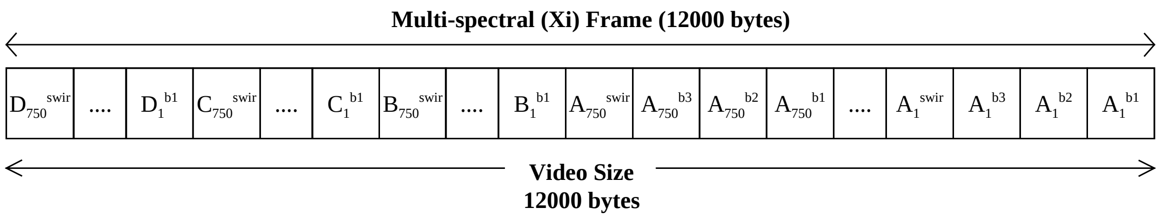
For further details about the fields of the above structure refer to DR8.

Sensor Scan data following the Auxiliary portion contains Video data organized in the following way.

- In the panchromatic (M) case there are two successive frames of 6000 Bytes each.
- In the Xi case there is one multi-spectral frame. A SWIR (infrared) 3000 pixels detector is added to the 3 visible bands of 3000 pixels each. The Video data structure is 12000 bytes long, organized as band interleaved.

The following picture depicts the concepts described above.



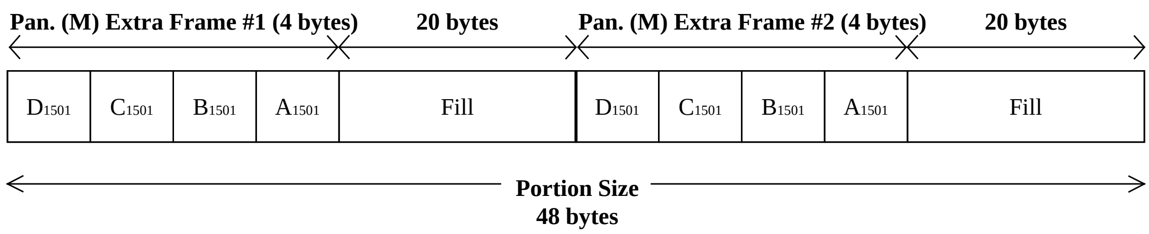


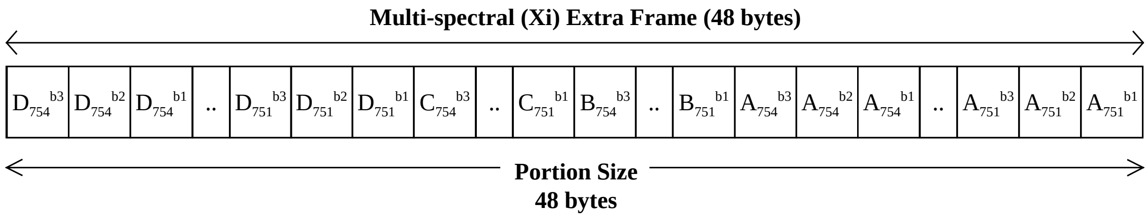
Frames of video data are arranged in order to be West-East oriented (pixel 1=East). To achieve this result the instrument barrettes are inverted.

Extra Pixels data portion following the Sensor Scan Data are organized in the following way.

- In the panchromatic (M) case there is one extra pixel per array (four each line). The remnant 40 bytes up to 48 are filled.
- In the Xi case there are four extra pixels each array (4 pixels x 4 arrays x 3 bands). The SWIR detector has not extra pixels. The portion is 48 bytes long, organized as band interleaved.

The following picture depicts the concepts described above.





2.4.3.8 X SAR "SENSOR ACQUIRED DATA"

The X SAR Sensor Data of one Data Take are stored consecutively in the file. The physical records have length equal to 1 Mbytes. This records fits a variable number of raw lines depending on the used Pulse Repetition Frequency (PRF).
This concept applies for SRL missions 1 & 2 as they can manage different PRF in the same data take. The 3rd mission (SRTM) has a fixed PRF (1674 Hz).
Note that different PRF means also different Format Line size (what in XSAR documents they call "Frame size").
The XSAR format (frame), as recorded by the sensor, is :

256 bits header + sar samples line of **variable size** (one among shown in the previous table)

See table below for PRF vs Frame Size :

PRF [Hz]	FRAME SIZE [bits]
1240	36288
1260	35712
1302	34560
1344	33480
1395	32256
1440	31248
1488	30240
1512	29760
1620	27776
1674	26880
1736	25920
1860	24192
1890	23808
1953	23040
2016	22320
2160	20832

For further detail about XSAR data format refer to [DR10].

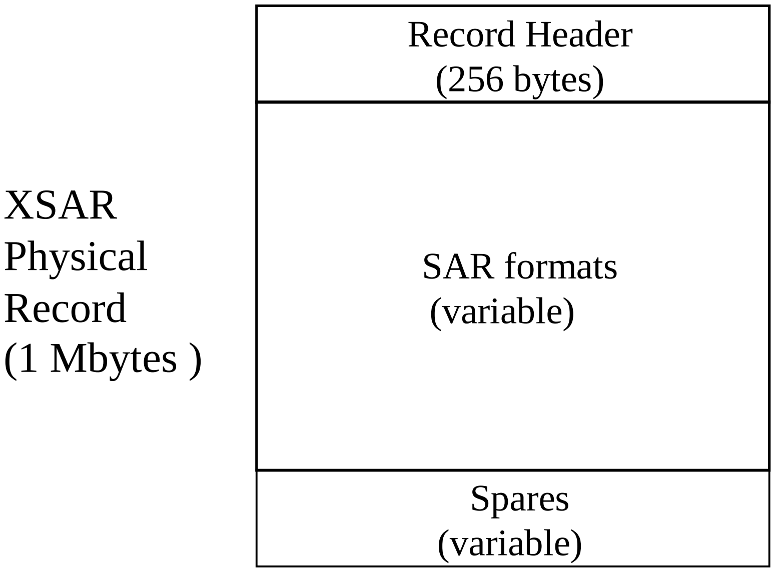
XSAR Sensor Data are stored in fixed length records, each containing a variable number of formats, depending on the frame size. We have chosen a fixed length (1 Mbytes) and a variable number of formats (from 231 to 402).
In the next table we show again PRF and Frame Size, with some other columns showing the number of formats per physical record and the corresponding physical record length (which is fixed , in fact, to 1 MBytes). In the last column we show the spares remaining in each record.

PRF [Hz]	FRAME SIZE [bits]	FRAMES PER RECORD	RECORD SIZE [Bytes]	SPARES PER RECORD [Bytes]
1240	36288	231	1048576	760
1260	35712	234	1048576	4000
1302	34560	242	1048576	3136
1344	33480	250	1048576	2326
1395	32256	260	1048576	256
1440	31248	268	1048576	1768

1488	30240	277	1048576	1516
1512	29760	281	1048576	3256
1620	27776	301	1048576	3504
1674	26880	312	1048576	256
1736	25920	323	1048576	2056
1860	24192	346	1048576	2272
1890	23808	352	1048576	1025
1953	23040	364	1048576	256
2016	22320	375	1048576	2326
2160	20832	402	1048576	1768

Data are transferred without any manipulation. No padding of lost lines is made.
A Block , as used in the Block Address Record, is directly referring to each physical record and therefore has the length of about 1 Megabytes.
Please note that record and block sizes must are set within the format in the Segment Description Records (2.4.5).
The place where one can read record sizes and number of formats per record (swath) is different with respect to the other sensors/satellites. We usually read these values from the User Header as they are fixed for all the Sensor Acquired Data file.
In this case it may vary so we have decoded to add, at the beginning of each physical record, an header having length of 256 bytes, taking advantage of the spares shown in the previous table.

The Physical records therefore have format :



The Record header has format :

Structure of a XSAR Record header

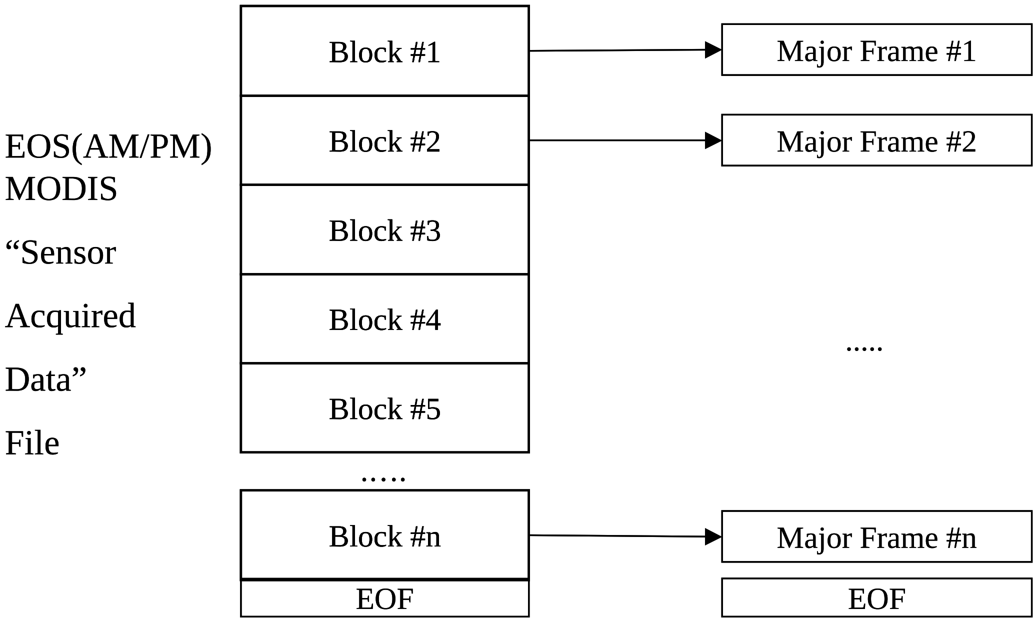
Field	Start	Stop	Bytes	Type	Description
Pulse_Repetition_Frequency	1	3	4	int	Pulse Repetition Frequency in Hertz
Raw_Line_Size	4	7	4	int	Raw line length in bytes
Frames_Per_Record	8	11	4	int	Number of Raw Lines (Frames) in this record
MET record start - DAY	12	15	4	int	MET DAY of start of this records
MET record start - HOURS	16	19	4	int	MET HOURS of start of this records
MET record start - MINUTES	20	23	4	int	MET MINUTES of start of this records
MET record start - SECONDS	24	27	4	int	MET SECONDS of start of this records
MET record stop - DAY	28	31	4	int	MET DAY of stop of this records
MET record stop - HOURS	32	36	4	int	MET HOURS of stop of this records
MET record stop - MINUTES	36	39	4	int	MET MINUTES of stop of this records
MET record stop - SECONDS	40	43	4	int	MET SECONDS of stop of this records
Reserved	44	255	212	char	reserved

Specific block characteristics must be read from the “Block Addresses Descriptor” file, as described in section “Data Retrieval Method” 2.3.1.

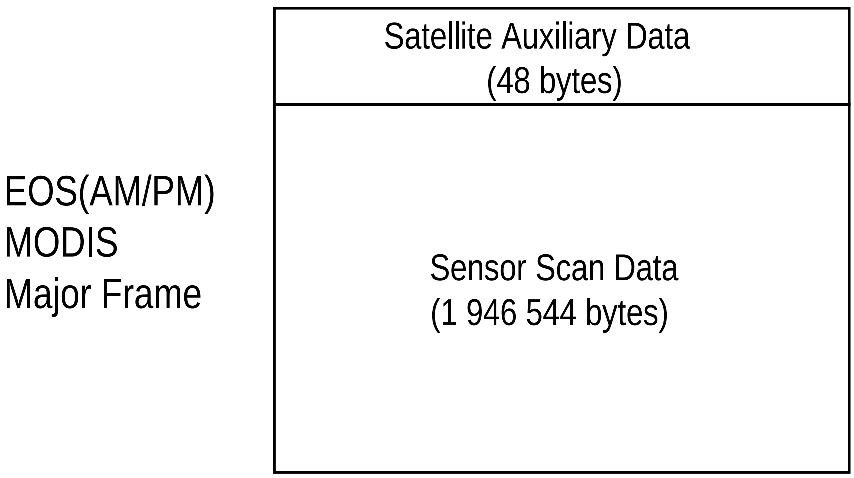
Note that for the XSAR third mission, named SRTM, the pulse repetition frequency is 1674 and therefore the spares indicated in the previous table are completely used for the Record Header.

2.4.3.9 EOS(AM/PM) MODIS "SENSOR ACQUIRED DATA"

The EOSAM/EOSPM MODIS Sensor Acquired Data are written on tape as follows.
The data corresponding to one pass are subdivided in Blocks having the same length of about 2 Megabytes. The real number of **Major Frames** per block must be read from the "User Header".
As an example, for EOSAM/EOSPM MODIS each block contains usually 1 complete Major Frame (or swath). However, this number can be changed according to output technical modifications, with the aim of improving the I/O speed.



Each Major Frame contains at least a complete scan (a 360 degree mirror turn around) of MODIS instrument and some other auxiliary information. It has the length of **1 946 592 bytes**, organised as follows:



Detailed description of a EOS(AM/PM) MODIS Major Frame structure follows.

MODIS MAJOR FRAME :

Field	Start	Stop	Bytes	Description
Satellite Auxiliary Data	1	48	48	Satellite Data Extracted from swath during transcription. Useful for swath processing.
Sensor Scan Data	49	1946592	1946544	MODIS sensor Scan Data

The Satellite Auxiliary Data structure reports information extracted from video data during the transcription procedure. It appears as follows.

Field	Start	Stop	Bytes	Type	Description
Year	1	4	4	u_int	Year of data acquisition
Day	5	8	4	u_int	# of days from the current year
Hours	9	12	4	u_int	# of hours from midnight
Minutes	13	16	4	u_int	# of minutes
Seconds	17	20	4	u_int	# of seconds
Milliseconds	21	24	4	u_int	# of milliseconds
Microseconds	25	28	4	u_int	# of microseconds
Scan type	29	32	4	u_int	Scan mode identifier: 0x00 Day scan mode 0x01 Night scan mode
APID	33	36	4	u_int	Application Process ID: 0x40 (64 decimal)
Fillers	37	40	4	u_int	Padding for structure alignment
Satellite Time	41	48	8	double	Swath time in milliseconds and part of them from the beginning of the year.

The Sensor Scan Data portion reports information in CCSDS packets form, extracted from the "Mission Data Zone" of downlinked CADUs. Each Sensor Scan Data portion contains:

- One complete (360 degrees) mirror turn around (a scan), if the MODIS sensor is operating in DAY mode;
- Three complete (360 degrees) mirror turn around (three scans) plus some spare bytes, if the MODIS sensor is operating in NIGHT mode;

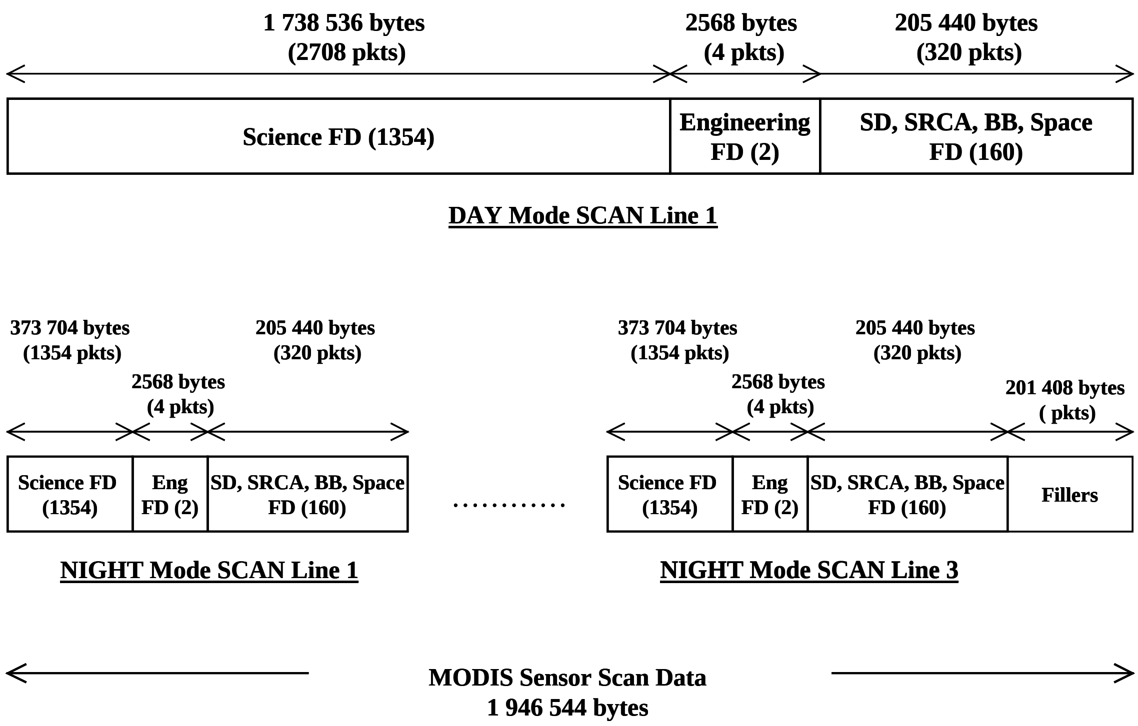
The table that follows explains in details the Sensor Scan Data portion of a Major Frame:

Field	Start	Stop	Bytes	Type	Description
					DAY mode
CCSDS Packet 1	1	642	642	u_char	First CCSDS packet of Science FD 1
CCSDS Packet 2	643	1284	642	u_char	Second CCSDS packet of Science FD 1
………….	……….	……….	642	u_char	……………………………………..
CCSDS Packet 2709	1738537	1739178	642	u_char	First CCSDS packet of Engineering group 1
………….	……….	……….	642	u_char	……………………………………..
CCSDS Packet 2713	1741105	1741746	642	u_char	First CCSDS packet of Calibration FD 1
………….	……….	……….	642	u_char	……………………………………..
CCSDS Packet 3032	1945903	1946544	642	u_char	Second CCSDS packet of Calibration FD 160
					NIGHT mode
CCSDS Packet 1	1	276	276	u_char	First CCSDS packet of Science FD 1 (Scan 1)
CCSDS Packet 2	277	552	276	u_char	First CCSDS packet of Science FD 2 (Scan 1)

………….	……….	……….	276	u_char	……………………………………..
CCSDS Packet 1355	373705	374346	642	u_char	First CCSDS packet of Engineering group 1 (Scan 1)
………….	……….	……….	642	u_char	……………………………………..
CCSDS Packet 1359	376273	376914	642	u_char	First CCSDS packet of Calibration FD 1 (Scan 1)
………….	……….	……….	642	u_char	……………………………………..
CCSDS Packet 1678	581071	581712	642	u_char	Second CCSDS packet of Calibration FD 160 (Scan 1)
CCSDS Packet 1	581713	581988	276	u_char	First CCSDS packet of Science FD 1 (Scan 2)
………….	……….	……….	……..	u_char	……………………………………..
CCSDS Packet 1678	1162783	1163424	642	u_char	Second CCSDS packet of Calibration FD 160 (Scan 2)
CCSDS Packet 1	1163425	1163700	276	u_char	First CCSDS packet of Science FD 1 (Scan 3)
………….	……….	……….	……..	u_char	……………………………………..
CCSDS Packet 1678	1744495	1745136	642	u_char	Second CCSDS packet of Calibration FD 160 (Scan 3)
Dummy	1745137	1946544	201408	u_char	Bytes filled with 0

The data unit of a scan is the CCSDS packet of 642 bytes for Science packets during day and of 276 bytes for Science packets during night. The CCSDS packets carrying Engineering and Calibration information have always the same size (642 bytes).
The CCSDS packets format can be found in *MODIS COMMAND, TELEMETRY, SCIENCE AND ENGINEERING DESCRIPTION* [DR 11].
The CCSDS stream is archived on DLT without any modification other then the bit error check and correction using the Reed-Solomon symbols attached at the VCDU level. The APID filtering is applied and only the packets carrying the MODIS APID are archived.

The Sensor Scan Data global structure can be represented graphically as follows:



2.4.4 "USER HEADER" FILE

General description

The "User Header" file contains the acquisition description as well as the logical and physical file structure. It contains all the parameters of the imaging sequence, the orbital parameters, the information about acquisition station, the description of the file structure and contents and all information necessary for further processing. It has the same structure for all satellites transcriptions. The file is divided into five logical sections:

1. Mission and Instrument Identification
2. Ground Stations and Transcription System Identification
3. Transcription Identification
4. Orbit and Acquisition Identification
5. Pointers to Tape Data Structure

Note: Codes assignments are provided in chapter 3

2.4.4.1 MISSION AND INSTRUMENT IDENTIFICATION

Field	Start	Stop	Bytes	Type	Description
Reserved	1	76	76	char	Reserved
Satellite_ID	77	78	2	short	Satellite Code
Mission_ID	79	80	2	short	Mission. Number
Instr_Type_ID	81	82	2	short	Instrument Type Code
Instr_Number	83	84	2	short	Instrument number (when applicable)
Transm_Channel	85	86	2	short	Transmission channel number (when applicable).

2.4.4.2 GROUND STATIONS AND TRANSCRIPTION SYSTEM IDENTIFICATION

Field	Start	Stop	Bytes	Type	Description
Station_ID	87	88	2	short	Acquisition Ground Station Code
Station_DT_ID	89	90	2	short	Data Transcription Ground Station Code:
Filler	91	92	2	char	Padding for structure alignment

2.4.4.3 TRANSCRIPTION IDENTIFICATION

Field	Start	Stop	Bytes	Type	Description
Input_HddrType	93	96	4	u_int	HDDR Code
Format_SyncTy	97	100	4	u_int	Format Synchroniser/Decommutation Code
Reserved	101	108	8	2*u_int	Reserved

2.4.4.4 ORBIT AND ACQUISITION IDENTIFICATION

Field	Start	Stop	Bytes	Type	Description
Track_Number	109	112	4	int	Track Number (when applicable)
Orbit_Number	113	116	4	int	Orbit number (when applicable)
Cycle_Number	117	120	4	int	Cycle number (when applicable)
Numb_of_Frames	121	124	4	int	Number of standard frames[1] (when applicable)
First_Frame	125	128	4	int	Num. of first standard frame[2] (when applicable)
Reserved	129	144	16	4*u_int	Reserved
First_OBC	145	148	4	u_int	1st On Board Counter (when applicable)
Last_OBC	149	152	4	u_int	Last On Board Counter (when applicable)
Acquis_Date	153	158	6	3*u_short	Acquisition Date in Y M D
Acquis_Day	159	160	2	u_short	Day in the year of the acquisition
Acquis_start	161	168	8	4* u_short	Start of acquisition. in Hours, Min, Sec, Millisec
Acquis_end	169	176	8	4*u_ short	End of acquisition. in Hours, Min, Sec, Millisec
Transcription_Date	177	182	6	3*u_short	Transcription Date in D M Y (WARNING: Year could be expressed in some tapes as years from 1900)
Transcription _Start	183	188	6	3*u_short	Transcription start in Hours, Min, Sec
Transcription _End	189	194	6	3*u_short	Transcription end in Hours, Min, Sec
Filler	195	196	2	char	Padding for stucture alignment

2.4.4.5 POINTERS TO TAPE DATA STRUCTURE

Field	Start	Stop	Bytes	Type	Description
Num. of segments	197	200	4	int	Number of segments (when applicable)
Loaded_Swath	201	204	4	int	Number of transcribed swaths
Swath_Size	205	208	4	int	Swath length (in bytes)
Swath_per_Block	209	212	4	int	Number of swaths per block
Nr_of_Blocks	213	216	4	int	Number of blocks
Physical_Address_1	217	220	4	long	User Header Address [3]
Physical_Address_2	221	224	4	long	Pass Id. Header Address [4]
Nr_of_Files	225	228	4	int	Number of files following the present
File_ID_1	229	232	4	int	File type identifier #1
File_Num_1	233	236	4	int	Number of physical records #1
Record_Length_1	237	240	4	int	Physical record length in bytes #1
Elem_Num_1	241	244	4	int	Number of logical element per record #1
Elem_Length_1	245	248	4	int	Logical element length in bytes #1
Filler_1	249	293	44	char	Spare #1
File_ID_2	294	297	4	int	File type identifier #2
File_Num_2	298	301	4	int	Number of physical records #2
Record_Length_2	302	305	4	int	Physical record length in bytes #2
Elem_Num_2	306	309	4	int	Number of logical element per record #2
Elem_Length_2	310	313	4	int	Logical element length in bytes #2
Filler_2	314	357	44	char	Spare #2
. . .					
File_ID_10	805	808	4	int	File type identifier #10
File_Num_10	809	812	4	int	Number of physical record #10

Record_Length_10	813	816	4	int	Physical record length in bytes #10
Elem_Num_10	817	820	4	int	Number of logical element per record #10
Elem_Length_10	821	824	4	int	Logical element length in bytes #10
Filler_10	825	868	44	char	Spare #10
Reserved	869	876	8	char	Reserved

A few words must be spent to explain the "Pointers to Tape Data" structure, previously shown as a table.
This structure is meaningful only in the "User Header" file (it is empty in "Pass Id. Header"). The last field of the first area (Nr_of_Files) indicates how many files follow the "User Header" inside the same Transcription Area (same passage).
This number has been taken as a parameter to allow the maximum flexibility.
In the Transcription Systems presently designed and installed the parameter "Nr_of_Files" can be 2 (LANDSAT), 3 (ERS SAR, SPOT HVR and HRVIR) or 4 (J-ERS SAR) (see figure depicting Transcription Area).
Nevertheless in the future other satellites could have different data structure and "Nr_of Files" could vary correspondingly up to 10.

The 10 blocks which follow "Nr_of Files" describes the files following the "User Header" in current "Transcription Area". Each of these blocks describes one file. The number of blocks actually filled is thus equal to "Nr_of_Files".

In the LANDSAT,EOSAM and J-ERS VNIR case there are 2 blocks filled, which correspond respectively to:

1. the "Segment Descriptor" file,
2. the "Block Addresses Descriptor" file.

In the EOSPM case there are 3 blocks filled, which correspond respectively to:

1. the "Segment Descriptor" file,
2. the "Payload Correction" file,
3. the "Block Addresses Descriptor" file.

In J-ERS case there are 4 blocks filled, which correspond respectively to:

1. the "Segment Descriptor" file,
2. the "Orbit Data" file,
3. the "Payload Correction" file,
4. the "Block Addresses Descriptor" file.

In ERS SAR, SPOT (HRV and HRVIR) and SHUTTLE XSAR and case there are 3 blocks filled, which correspond respectively to:

1. the "Segment Descriptor" file,
2. the "Orbit Data" file,
3. the "Block Addresses Descriptor" file.

2.4.5 "SEGMENT DESCRIPTOR" FILE

This file contains the descriptions of all the segments in which the satellite pass has been divided.
Each segment is described by one structure. The "Segment Descriptor Structure" is thus repeated as many times as the number of segments recorded in the "Sensor Acquired Data" file.
Each segment is completely described by the following fields and contains all information to address it within the video data records.
The number of items and the length are specified into the "User Header file - Pointer to Tape Data Structure Description Section" (described in previous paragraph).

2.4.5.1 SEGMENT AND ACQUISITION IDENTIFICATION

Field	Start	Stop	Bytes	Type	Description
Acquis_Date	1	6	6	3*short	Acquisition Date of the Sat. Pass in Y, M, D
Acquis_Day	7	8	2	short	Day in the year of the acquisition
Segment_Start	9	16	8	4*short	Start of Segment in Hours, Min, Sec, Millisec
Segment_End	17	24	8	4*short	End of Segment in Hours, Min, Sec, Millisec
Loaded_Swath	25	28	4	int	Nr. of lines loaded on tape for this segment
First_Swath	29	32	4	int	First swath of the segment
Last_Swath	33	36	4	int	Last swath of the segment
Lost_Swath	37	40	4	int	Lost swaths of the segment
First_Frame	41	44	4	int	First frame of the segment (when applicable)
Last_Frame	45	48	4	int	Last frame of the segment (when applicable)
First_OBC	49	52	4	int	First On Board Counter or TSID (when applicable)
Last_OBC	53	56	4	int	Last On Board Counter or TSID (when applicable)

2.4.5.2 POINTERS TO TAPE DATA STRUCTURE

Field	Start	Stop	Bytes	Type	Description
Starting_Address	57	60	4	long	Pass Id. Header file Address
Swath_Size	61	64	4	int	Swath length (in bytes) ([9])
Swath_per_Block	65	68	4	int	Number of swaths per block ([10])
Nr_of_Blocks	69	72	4	int	Number of blocks ([11])
Formats_Per_Swath	73	76	4	int	Number of formats per swath ([12])
Filler	77	128	52	char	Filler

[9] This field is filled only if its value changed between different Segments. If it does not change the valid value for this field must be read in the User Header (Pointers to Tape Data Structure)

[10] This field is filled only if its value changed between different Segments. If it does not change the valid value for this field must be read in the User Header (Pointers to Tape Data Structure)

[11] This field is filled only if its value changed between different Segments. If it does not change the valid value for this field must be read in the User Header (Pointers to Tape Data Structure)

[12] This field is filled only if its value changed between different Segments. If it does not change the valid value for this field must be read in the User Header (Pointers to Tape Data Structure)

2.4.6 "ORBIT DATA" FILE

2.4.6.1 ERS, J-ERS SAR CASE

This structure is composed by a series of blocks containing information about position, velocity of satellite and times closest to start acquisition.
In the ERS and J-ERS SAR case there is only one "Orbital Data Record" for each pass, i.e. for each Transcription Area.
The State Vector is referenced to the inertial co-ordinate system called ECITOD (Earth Centred Inertial True Of Date).
The "Orbital Data" record has the following structure.

Field	Start	Stop	Bytes	Type	Description
State_Vector_X	1	8	8	double	X component of satellite position
State_Vector_Y	9	16	8	double	Y component of satellite position
State_Vector_Z	17	24	8	double	Z component of satellite position
St_Vect_X_Vel	25	32	8	double	Vx component of sat. velocity
St_Vect_Y_Vel	33	40	8	double	Vy component of sat. velocity
St_Vect_Z_Vel	41	48	8	double	Vz component of sat. velocity
AscNodeUTC	49	73	25	char	UTC epoch of sat. position
ReferenceUTC	74	98	25	char	UTC reference time
Filler	99	100	2	char	Padding for structure alignment
SatBinTime	101	104	4	u_long	Satellite reference binary time
ClockStepLength	105	108	4	see footnote	See footnote ([13])
DBType	109	112	4	int	0 = predicted, 1 = restituted

[13] The field is satellite dependent, as follows:

Satellite/Sensor	Type	Description
J-ERS SAR	signed long	Time correction in millisecond
ERS SAR	unsigned long	Step length of clock in nanoseconds

2.4.6.2 SPOT 1-2-3 CASE

Here follows the structure of SPOT 1-2-3 orbital data.
They are described in a top down way, i.e. starting from high level structure and going down in further detail.
The C structure are provided in the following and explained with comments.
The overall structure, named SpotInfo, is the following:

```
typedef struct {
      unsigned int                a;       /* reserved */
      unsigned int                b;       /* reserved */

      ReservRequest               data1 ;  /* SPOT card 1 */
      RespToReserv                data2 ;  /* SPOT card 2 */
      ProgRequest                 data3 ;  /* SPOT card 3 */
      RespToProg                  data4 ;  /* SPOT card 4 */
      AntennaPtData               data5 ;  /* SPOT card 5 */
      ImgSchedPlan                data6 ;  /* SPOT card 6 */
      PostPassEphem               data7 ;  /* SPOT card 7 */
} SpotInfo;
```

Here follows the structure of the cards:

2.4.6.2.1 CARD 1 (ReservRequest)

Field	Start	Stop	Bytes	Type	Description
filled	1	3	4	Boolean	Reserved
day	4	7	4	u_int	day number of year
mission	8	11	4	u_int	mission number
orbit	12	15	4	u_int	orbit number
req_type	16	19	4	int	request type: 1=default, 2=no-default
begin_lat	20	23	4	float	latitude of beginning; given in degrees (-90 to 90)
end_lat	24	27	4	float	latitude of end; given in degrees (-90 to 90)

2.4.6.2.2 CARD 2 (RespToReserv)

Field	Start	Stop	Bytes	Type	Description
filled	1	3	4	Boolean	Reserved
day	4	7	4	u_int	day number of year
mission	8	11	4	u_int	mission number
orbit	12	15	4	u_int	orbit number
diagnostic	16	19	4	int	diagnostic: 0 = refused 10 = accepted in part 20 = fully accepte
begin_lat	20	23	4	float	latitude of beginning; given in degrees (-90 to 90)
end_lat	24	27	4	float	latitude of end; given in degrees (-90 to 90)

2.4.6.2.3 CARD 3 (ProgRequest)

Field	Start	Stop	Bytes	Type	Description
filled	1	3	4	Boolean	Reserved
day	4	7	4	u_int	day number of year
mission	8	11	4	u_int	mission number
orbit	12	15	4	u_int	orbit number
prog_type	16	19	4	int	programming type: 1=HRV 2=COUPLED HRV 3=INDEPENDENT HRVS
segments_nb	20	23	4	u_int	number of segments
xs_gain 1	24	27	4	int	gains for multispectral : 1 0=standard 1=high 2=low
xs_gain 2	28	31	4	int	gains for multispectral : 2 0=standard 1=high 2=low
xs_gain 3	32	35	4	int	gains for multispectral : 3 0=standard 1=high 2=low
pa_gain	36	39	4	u_int	
segment struct	40	1239	1200	struct	array of struct segments of [50] elements

2.4.6.2.3.1 Segment structure

Field	Start	Stop	Bytes	Type	Description
spec_mode	1	3	4	int	spectral mode : 1=PA 4=PAXS 2=PADPCM 5= PADPCMXS 3=XS
cen_lon	4	7	4	float	degrees (0 to 360)
cen_lat	8	11	4	float	degrees (0 to 360)
spec_mode2	12	15	4	int	These last 3 fields are used if program. type is INDEPENDENT HRVS (see CARD 3 above)
cen_lon2	16	19	4	float	
cen_lat2	20	23	4	float	

2.4.6.2.4 CARD 4 (RespToProg)

Field	Start	Stop	Bytes	Type	Description
filled	1	3	4	Boolean	Reserved
day	4	7	4	u_int	day number of year
mission	8	11	4	u_int	mission number
orbit	12	15	4	u_int	orbit number
diagnostic	16	19	4	int	diagnostic: 0 = refused

					10=Accepted 20=Centering point unacceptable 30=Too many segments 40=Specular reflection 50=Centering point outside reserved slot 60=Segment too short (<9) 70=Wrong payload configuration
swath_nb	20	23	4	u_int	number of swaths
swath struct	24	263	240	struct	swath structure of [10] elements

2.4.6.2.4.1 Swath structure

Field	Start	Stop	Bytes	Type	Description
spec_mode	1	3	4	int	spectral mode : 1=PA 4=PAXS 2=PADPCM 5= PADPCMXS 3=XS
begin_lon	4	7	4	float	longitude of the beginning
begin_lat	8	11	4	float	latitude of the beginning
end_lon	12	15	4	float	longitude of the end
end_lat	16	19	4	float	latitude of the end
angle	20	23	4	float	reserved

2.4.6.2.5 CARD 5 (AntennaPtData)

Field	Start	Stop	Bytes	Type	Description
filled	1	3	4	Boolean	Reserved
day	4	7	4	u_int	day number of year
mission	8	11	4	u_int	mission number
orbit	12	15	4	u_int	orbit number
jday	16	19	4	u_int	julian day date
filler	20	23	4	int	padding bytes
begin_sec	24	31	8	double	begin seconds (jd)
end_sec	32	39	8	double	end seconds (jd)
data	40	43	4	int	antenna-pointing data: 0=osculating case: oelem[6] is filled, epoint[50] is all zero 1=ephemeris case: epoint[50] is filled, oelem[6] is zero
filler	44	47	4	int	padding bytes
osculanting elements	48	191	24*6=144	struct	OsculatingElem struct of [6] elements
epoint_nb	192	195	4	u_int	ephemeris point number
jd_first_pt	196	199	4	u_int	julian date of the first line of the segment
sec_first_pt	200	203	4	u_int	seconds of the first line of the segment
filler	204	207	4	int	padding bytes
ephemeris point	208	3007	56*50=2800	struct	EphemerisPoint struct of [50] elements

2.4.6.2.5.1 Osculating Element structure

Field	Start	Stop	Bytes	Type	Description
otype	1	3	4	char	otype[0]=code number otype[1]=code number otype[2]=blank otype[3]=type of osculating A= semi-major axis E= eccentricity I= orbit inclination PO= perigee argument M= mean anomaly
Filler	4	7	4	char	Padding for structure alignment
value	8	15	8	double	value of osculating element
unit	16	18	3	char	measurement unit
Filler	19	19	1	char	Padding for structure alignment
checksum	20	24	4	float	checksum (module 10 sum of digits of each osculating element)

2.4.6.2.5.2 Ephemeris Point structure

Field	Start	Stop	Bytes	Type	Description
point_id	1	3	4	u_int	point identifier
Filler	4	7	4	char	Padding for structure alignment
x_km	8	15	8	double	x position in km
y_km	16	23	8	double	y position in km
z_km	24	31	8	double	z position in km
x_km_s	32	39	8	double	x velocity in km/s
y_km_s	40	47	8	double	y velocity in km/s
z_km_s	48	55	8	double	z velocity in km/s

2.4.6.2.6 CARD 6 (ImgSchedPlan)

Field	Start	Stop	Bytes	Type	Description
filled	1	3	4	Boolean	Reserved
day	4	7	4	u_int	day number of year
mission	8	11	4	u_int	mission number
orbit	12	15	4	u_int	orbit number
diagnostic	16	19	4	int	diagnostic 10=station programming 20=default programming 30=no programming
chan1_segm_nb	20	23	4	u_int	channel 1 segment number
chan2_segm_nb	24	27	4	u_int	channel 2 segment number
obt	28	31	4	u_long	on board time of pass
To	32	35	4	u_int	julian-day date
to	36	39	4	float	time-of-day in decimal seconds with resol. of 1ms
a	40	43	4	float	period of on-board counter
co	44	47	4	u_int	count of on-board counter
chan1_segm	48	2847	2800	struct	ChannelSegm array of [50] elements

chan2_segm	2848	5647	2800	struct	ChannelSegm array of [50] elements

2.4.6.2.6.1 Channel Segment structure

Field	Start	Stop	Bytes	Type	Description
chan_id	1	3	4	int	channel identifier: 1 or 2
begin_counter	4	7	4	u_int	format counter of the beginning of segment
end_counter	8	11	4	u_int	format counter of the end of segment
hrv_id	12	15	4	int	hrv identifier
spec_mode	16	16	1	char	X=XS, P=PA, D=PA(DPCM)
segment	17	17	1	char	V=imaging
Filler	18	19	2	char	Padding for structure alignment
test_indicator	20	23	4	int	always 0
pa_gain	24	27	4	int	PA gain value
xs_gain	28	31	4	int	XS gain values
xs_gain	32	35	4	int	XS gain values
xs_gain	36	39	4	int	XS gain values
ssm_angle	40	43	4	int	SSM pointing angle (in steps, from 32 to 52)
status	44	47	4	int	SSM status indicator: 0=normal, 1 degraded
validity	48	51	4	int	always 1
ref_pos	52	55	4	int	refocus position (in steps, from 32 to 52)

2.4.6.2.7 CARD 7 (PostPassEphem)

Field	Start	Stop	Bytes	Type	Description
filled	1	3	4	Boolean	Reserved
day	4	7	4	u_int	day number of year
mission	8	11	4	u_int	mission number
orbit	12	15	4	u_int	orbit number
To	16	19	4	u_int	julian-day date
to	20	23	4	float	time-of-day in decimal seconds with resol. of 1ms
a	24	27	4	float	period of on-board counter
co	28	31	4	u_long	count of on-board counter
epoint_nb	32	35	4	u_int	ephemeris point number of the first line of segment
jd_first_pt	36	39	4	u_int	julian date of the first line of segment
sec_first_pt	40	43	4	u_int	seconds of day of the first line of segment
Filler	44	47	4	char	Padding for structure alignment
Ephemeris Point	48	2847	56*50=2800	struct	EphemerisPoint array of [50] elements

2.4.6.3 SPOT 4 CASE

Here follows the structure of SPOT 4 Orbital Data file.
The satellite working mode may programmed, depending on Spot Image Planning, in the old SPOT 1-2-3 mode and in the new SPOT 4 mode. In order to manage this double mode, the “Orbit Data File” is grown in size and a dedicated structure has been added, written on the DLT after the Spot_Info structure. The file size and record number has however to be read in the “User Header” file - Pointers to Tape Data Structure field.
The following table shows the difference in file content when one of the two working modes is used.

	Spot_Info structure	Spot_4_Info structure	
SPOT 1-2-3 working mode	Filled	Empty	
SPOT 4 working mode	Filled	Filled	

The structure named Spot_Info identical to the SPOT 1-2-3 structure described in the previous chapter, while the structure Spot_4_Info is dedicated to the new working mode. It is described in a top down way, i.e. starting from high level structure and going down in further detail.
The C structure are provided in the following and explained with comments. For details and explanations see DR9.

The overall structure, named Spot_4_Info, is the following:

```
typedef struct {
        Boolean             Filled;           /*    If FALSE the
structure is empty */
        Boolean             Madras_Flag;      /* reserved */
        Boolean             Diode_Flag;       /* reserved */
        Sequence4Plan       data0;            /* SPOT 4 card 0 */
        Sequence4Plan       data1;            /* SPOT 4 card 1 */
        Resp4ToReserv       data2;            /* SPOT 4 card 2 */
        Prog4Description    data3;            /* SPOT 4 card 3 */
        Resp4ToProg         data4;            /* SPOT 4 card 4 */
        Antenna4PtData      data5;            /* SPOT 4 card 5 */
        Img4SchedPlan       data6;            /* SPOT 4 card 6 */
          Img4DRSSchedPlan  data8;            /* SPOT 4 card 8 */
} Spot_4_Info;
```

Here follows the structure of the cards:

<u>SPOT 4 Card 0 and Card 1</u>

```
typedef struct {
        Boolean        filled;                /* reserved */
        unsigned int                          segments_nb;      /* number of segments  */
        Sequence4PlanSegment   segments[50];  /* The structure Sequence4PlanSegment
                                              is described after */
} Sequence4Plan;                              /* 00 */ /* 01 */
```

```
typedef struct {
        unsigned int    day;            /* day number of year */

        char[3]         mission;        /* mission number */
        unsigned int    orbit;          /* orbit number */
        float           begin_lat;      /* latitude of beginning */
        float           end_lat;        /* latitude of end */
} Sequence4PlanSegment;
```

NOTE:

- The day number in the year is the day (in UTC) when the satellite enters the 0° elevation circle of the station.
- The mission field reports the three characters 'S4\0'.
- The begin and end latitude are given in degrees, between –90 and +90.
- The begin, center and end longitude are given in degrees eastwards, between 0 and 360.

Concerning the above fields, the same notation will be used in the rest of the document, except different specifications.

SPOT 4 Card 2

```
typedef struct {
    Boolean             filled;         /* reserved */
    unsigned int        nb_slots;       /* number of slots filled */
    Resp4ToReservSlot   slots[50];      /* The structure Resp4ToReservSlot is described after */
} Resp4ToReserve;                                               /* 02 */

typedef struct {
        unsigned int    day;            /* day number of year */
        char[3]         mission;        /* mission number */
        unsigned int    orbit;          /* orbit number */
        char            encoding;       /*      D = default encoding
                                                C = no encoded (clear)
                                                T = test encoding */
        float           begin_lat;      /* latitude of beginning */
        float           end_lat;        /* latitude of end */
        double          begin_sec;      /* seconds of the beginning */
        double          end_sec;        /* seconds of the end */
} Resp4ToReservSlot;
```

SPOT 4 Card 3

```
typedef struct {
        Boolean         filled;         /* reserved */
        unsigned int    day;            /* day number of year */
        char[3]         mission;        /* mission number */
        unsigned int    orbit;          /* orbit number */
        unsigned int    segments_nb;    /* number of segments */
        Spot4Segment    segments[50];   /* The structure Spot4Segment is described after */
} Prog4Description;                                             /* 03 */
```

```
typedef struct {
        unsigned int        coupling_mode;        /*    Coupling mode of segments:
                                                        10 = BI (Independent)
                                                        20 = DO (Double)
                                                        30 = TW (Twin)     */
        unsigned int        HRVIR_instrument;     /*    Instrument used:
                                                        10 = H0 (when coupling TW)
                                                        20 = H1
                                                        30 = H2            */
        unsigned int        spectral_mode;        /*    Instrument spectral mode:
                                                        10 = M
                                                        20 = MI   (when coupling DO)
                                                        30 = XI            */
        float               center_lon;           /* longitude of center */
        float               center_lat;           /* latitude of center */
        char                Xi_gains[4];          /*  gains for multi-spectral case:
                                                        H   = high
                                                        M   = medium
                                                        L   = low (default)
                                                        A value from 1 to 6     */
        char                M_gain[4];            /*  gain for panchromatic case */
        char                Xi_gains_2[4];        /*  gains for multi-spectral case (when coupling TW) */
        char                M_gain_2[4];          /*  gain for panchromatic case (when
coupling TW) */
} Spot4Segment;
```

SPOT 4 Card 4

```
typedef struct {
    Boolean             filled;           /* reserved */
    unsigned int        day;              /* day number of year  */
    char[3]             mission;          /* mission number      */
    unsigned int        orbit;            /* orbit number        */
    unsigned int        diagnostic;       /* diagnostic:
                10=Accepted
                20=Centering point unacceptable
                30=Too many segments
                40=Specular reflection
```

```
                    50=Centering point outside reserved slot

                    60=Segment too short (<9)

                    70=Wrong payload configuration      */
    unsigned int            swath_nb;          /* number of swaths accepted or rejected   */
    Spot4Swath              swaths[10];        /* The structure Spot4Swath is described after */
} Resp4ToProg;                                                  /* 04 */


typedef struct {
                                    unsigned int            coupling_mode;
        /*  Coupling mode of segments:

                                                            10 = BI
(Independent)

                                                            20 = DO (Double)

                                                            30 = TW (Twin)      */
                                    unsigned int            HRVIR_instrument;
    /*  Instrument used:

                                                            10 = H0 (when
coupling is TW)

                                                            20 = H1

                                                            30 = H2
                    */
                                    unsigned int            spectral_mode;
        /*  Instrument spectral mode:

                                                            10 = M

                                                            20 = MI    (when
coupling is DO)

                                                            30 = XI
                    */
    float                       begin_lon;                      /* longitude
of the beginning */
    float                       begin_lat;                  /* latitude of the
beginning */
    float                       end_lon;                    /* longitude of the
end */
    float                       end_lat;                        /* latitude
of the end */
    float                       angle;                          /* viewing
angle */
} Spot4Swath;
```

SPOT 4 Card 5

```
typedef struct {
    Boolean             filled;         /* reserved */
    unsigned int        day;            /* day number of year */
    char[3]             mission;        /* mission number */
    unsigned int        orbit;          /* orbit number */
    unsigned int        jday;           /* julian day date */
    double              begin_sec;      /* begin seconds (jd) */
    double              end_sec;        /* end seconds (jd) */
    int                 data;           /* antenna-pointing data:
                        0=osculating case: oelem[6] is filled, epoint[50] is all zero
                        1=ephemeris case: epoint[50] is filled, oelem[6] is zero, see after */
    OsculatingElem      oelems[6];      /* The structure OsculatingElem is described after*/
    unsigned int        epoint_nb;      /* ephemeris point number */
    unsigned int        jd_first_pt;    /* julian date of the first line of the segment */
    unsigned int        sec_first_pt;   /* seconds of the first line of the segment */
    EphemerisPoint      epoint[50];     /* The structure EphemerisPoint is described after */
} Antenna4PtData;                       /* 05 */

typedef struct {
    char     otype[4];      /*  otype[0]=code number
                                otype[1]=code number
                                otype[2]=blank
                                otype[3]=type of osculating element:
                                A=   semi-major axis
                                E=   eccentricity
                                I=   orbit inclination
                                PO=  perigee argument
                                GO=  east longitude of ascending node minus the sideral time
                                     corresponding to the beginning of imaging slot
                                M=   mean anomaly   */
    double   value;         /* value of osculating element */
    char     unit[3];       /* measurement unit */
    int      checksum;      /* checksum (module 10 sum of digits of each osculating element) */
} OsculatingElem;

typedef struct {
    unsigned int   point_id;  /* point identifier */
    double         x_km;      /* x position in km */
    double         y_km;      /* y position in km */
    double         z_km;      /* z position in km */
    double         x_km_s;    /* x velocity in km/s */
    double         y_km_s;    /* y velocity in km/s */
    double         z_km_s;    /* z velocity in km/s */
} EphemerisPoint;
```

SPOT 4 Card 6

```
typedef struct {
    Boolean          filled;              /* reserved */
    unsigned int     day;                 /* day number of year */
    char[3]          mission;             /* mission number */
    unsigned int     orbit;               /* orbit number */
    unsigned int     diagnostic;          /* station programming:

                     10 = programming accepted

                     20 = default programming applied

                     30 = no programming    */
    unsigned int     chan1_segm_nb;       /* channel 1 segment number */
    unsigned int     chan2_segm_nb;       /* channel 2 segment number */
    Channel4Segm     chan1_segm[50];  /* the structure Channel4Segm is described after */
    Channel4Segm     chan2_segm[50];  /* the structure Channel4Segm is described after */
} Img4SchedPlan;                                                          /* 06 */

typedef struct {
        char[3]          chan_id;                  /* channel identifier: C1 or C2
*/
        unsigned int     segment_counter;     /* counter of segments in the pass */
                     unsigned int     HRVIR_instrument;     /*     Instrument used:

                                                                  10 = H1

                                                                  20 = H2

                     */
                                          unsigned int          spectral_mode;
        /*      Instrument spectral mode:

                                                                  10 = M

                                                                  20 = XI

                     */
        float                          begin_lon;                      /* longitude
of the beginning */
        float                          begin_lat;                 /* latitude of the
beginning */
        float                          end_lon;                   /* longitude of the
end */
        float                          end_lat;                        /* latitude
of the end */
        float                          angle;                          /* viewing
angle */
                     double                 begin_sec;                        /*
seconds of the beginning */
                     double                 end_sec;                       /* seconds
of the end */
} Channel4Segm;
```

SPOT 4 Card 8

At the time of this edition, the card has not been defined, as reported in DR9.

2.4.6.4 X SAR CASE

In the X SAR case,the "Orbital Data Record" is loaded according to the AODA file, containing sensor position, velocity and attitude. Baseline between the two antennas for SRTM mission is also reported.
The State Vector is referenced to the inertial co-ordinate system called ECITOD (Earth Centred Inertial True Of Date).
The "Orbital Data" record has the following structure.

Field	Start	Stop	Bytes	Type	Description
MET_Seconds	1	8	8	double	Secs from Mission Start to which next params refer to.
State_Vector_X	9	16	8	double	X component of satellite position [meters]
State_Vector_Y	17	24	8	double	Y component of satellite position [meters]
State_Vector_Z	25	32	8	double	Z component of satellite position [meters]
St_Vect_X_Vel	33	40	8	double	Vx component of sat. velocity [meter/sec]
St_Vect_Y_Vel	41	48	8	double	Vy component of sat. velocity[meter/sec]
St_Vect_Z_Vel	49	56	8	double	Vz component of sat. velocity[meter/sec]
Baseline_X	57	64	8	double	X component of Baseline between the two antennas (for SRTM only)
Baseline_Y	65	72	8	double	Y component of Baseline between the two antennas (for SRTM only)
Baseline_Z	73	80	8	double	Z component of Baseline between the two antennas (for SRTM only)
Attitude _X	81	88	8	double	X component of Attitude
Attitude_Y	89	96	8	double	Y component of Attitude
Attitude Z	97	104	8	double	Z component of Attitude
Delta_State_Vector_X	105	112	8	double	X component of the Estimated Error on satellite position [meters]
Delta_State_Vector_Y	113	120	8	double	Y component of Estimated Error on satellite position [meters]
Delta_State_Vector_Z	121	128	8	double	Z component of Estimated Error on satellite position [meters]
Delta_St_Vect_X_Vel	129	136	8	double	Vx component of Estimated Error on sat. velocity [meter/sec]
Delta_St_Vect_Y_Vel	137	144	8	double	Vy component of Estimated Error on sat. velocity[meter/sec]
Delta_St_Vect_Z_Vel	145	152	8	double	Vz component of Estimated Error on sat. velocity[meter/sec]
Delta_Baseline_X	153	160	8	double	X component of Estimated Error on Baseline between the two antennas (for SRTM only)
Delta_Baseline_Y	161	168	8	double	Y component of Estimated Error on Baseline between the two antennas (for SRTM only)
Delta_Baseline_Z	169	176	8	double	Z component of Estimated Error on Baseline between the two antennas (for SRTM only)
Attitude _X	177	184	8	double	X component of Attitude
Attitude_Y	185	192	8	double	Y component of Attitude
Attitude Z	193	200	8	double	Z component of Attitude

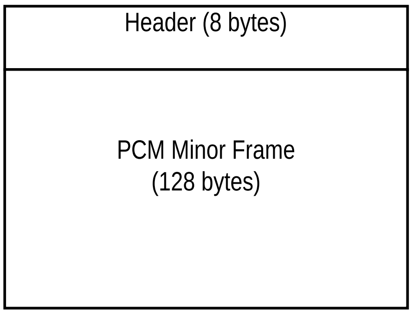
The number of records present in the Orbit Data File is written in the User Header (Pointers to Tape Data Structure) (2.4.4.5). It is supposed to be always one but now-a-days it is not yet specified.

2.4.7 "PAYLOAD CORRECTION" FILE

2.4.7.1 J-ERS SAR CASE

"Payload Correction" file is reported here for J-ERS SAR transcriptions. In this case, PCM data are extracted and transcribed onto DLT tape in a separate file. Of course, they are still present multiplexed in data lines. These data are transcribed in their emission form, as before multiplexing procedure. They are synchronised and formatted in Minor Frames and an header is added to form a record as follows.
PCM Minor Frames are formatted at a rate of two per second. Up to about 1500 logical records can so be found in a "Payload Correction" file (for 12 minutes of maximum pass duration). However, records number and length of present file must be read from "User Header" file at section "Pointers to Tape Data Structure", where detailed description of files following the "User Header" is reported.

Record #i



The Header reports information about the data continuity and has the following form.
The structure and field content of a PCM Minor Frame can be found in DR5.

Field	Start	Stop	Bytes	Type	Description
Complete Flag	1	1	1	u_char	If found: 0x00 = Complete Minor Frame 0xFF = Incomplete MF
Loss Start	2	2	1	u_char	First byte of MF set to 0 (set if incomplete)
Lost Lines	3	4	2	u_short	# of lost lines in video data during extraction (set if incomplete)
Line Counter	5	8	4	int	Line Counter referred to video line containing first sync bit of present MF. Masked with 0x00FFFFFF

2.4.7.2 EOSPM MODIS CASE

"Payload Correction" file is reported here for EOSPM MODIS transcriptions. In this case, GBAD 1 seconds packets are extracted and transcribed onto DLT tape in a separate file. These packets are **not** found in the Video Data.
GBAD data are transcribed in their emission form, as extracted by de-multiplexing procedure. They are synchronised and formatted in Instrument Source Packets without any header added.
GBAD packets are formatted at a rate of one per second. Records number and length of present file must be read from "User Header" file at section "Pointers to Tape Data Structure", where detailed description of files following the "User Header" is reported.

2.4.8 "BLOCK ADDRESSES DESCRIPTOR" FILE

This file contains the description of all the blocks in which the pass has been divided and written on tape. It is composed by a variable number of identical units. Each unit describes completely one satellite data block and contains all information to address any segment or any frame within the "Sensor Acquired Data" file. It contains block number of the video data block, the starting time of the first satellite format in the block and the number of swaths contained.
The structure is repeated as many times as the number of blocks recorded in the file.
The number of items and the length are specified into the "User Header file - Pointer to Tape Data Structure Description Section" (Block Addresses section).
This is the last file of the Transcription Area and is always followed by a Statistic File.

Field	Start	Stop	Bytes	Type	Description
Video_Start_Addr	1	4	4	long	Block number of the video data block
Satellite_Time	5	12	8	int	Time of current block ([14])
Delta_Time	13	16	4	int	Time distance between start of acquisition (first block of the file) and current block start ([15])
Swath_per_Block	17	20	4	u_int	Number of swaths per block
Filler	21	32	12	char	Available fields

[14] The time is expressed in sensor dependent mode, as follows:

Satellite/Sensor	Type	Description
LANDSAT ETM+, TM, MSS	Double	Milliseconds from midnight of first block swath
EOS(AM/PM) MODIS	Double	Milliseconds from midnight of first block swath
J-ERS VNIR	Double	Milliseconds from midnight of first block swath
J-ERS SAR	2*u_int	Sat_Time[0] = Milliseconds from midnight of first line in current block Sat_Time[1] = Milliseconds from midnight of last line in current block
ERS	2*u_int	Sat_Time[0] = Binary Counter of first line in current block Sat_Time[1] = Binary Counter of last line in current block
SPOT 1-2-3 HRV	2*u_int	Sat_Time[0] = First valid "On board counter" (see Auxiliary) of the pass Sat_Time[1] = Format counter of first format in the current block
SPOT 4 HRVIR	2*u_int	Sat_Time[0] = First valid "On board counter" (see Auxiliary) of the pass Sat_Time[1] = Format counter of first format in the current block
SHUTTLE(XSAR)	2*(4*u_char)	Sat_Time[0] = Start MET time [days+hours+sec+msec] Sat_Time[1] = Stop MET time [days+hours+sec+msec]

[15] The delta time is expressed in sensor dependent mode, as follows:

Satellite/Sensor	Type	Description
LANDSAT ETM+, TM, MSS	u_int	Milliseconds
EOS(AM/PM) MODIS	u_int	Milliseconds
J-ERS (SAR, VNIR)	u-int	Milliseconds
ERS SAR	u_int	Format counter of first format in current block
SPOT 1-2-3-4	u_int	Format counter difference
SHUTTLE(XSAR)	u_int	Milliseconds

2.5 STATISTICS FILE

The Statistics file always follows the Transcription Area in the DLT and constitutes a sort of summary of the passes contained in the DLT.
The Statistics File following the last Transcription Area of the cassette constitutes a directory, with information on all the transcribed passes.
This structure allows an easy positioning of the tape on the requested pass. To achieve this goal, when the tape inspection is performed, the end of tape is reached without reading any data. Then the control jumps one file back (at the beginning of the last Statistics file). Reading this file the content of the whole DLT can be known.
The first record is empty. The successive records are structured according the table below; each of them points to the successive Transcription Area stored on the cassette.
Each record of this file has the following structure.

Note: Codes assignments are provided at the end of paragraph 3.4.5.

Field	Start	Stop	Bytes	Type	Description
Reserved	1	4	4	char	Reserved
Satellite_ID	5	6	2	u_short	Satellite Code
Mission_ID	7	8	2	u_short	Mission. Number
Instr_Type_ID	9	10	2	u_short	Instrument Type Code
Reserved	11	14	4	2*u_short	Reserved
Station_ID	15	16	2	u_short	Acquisition Ground Station Code
Reserved	17	18	2	short	Reserved
Filler	19	20	2	char	Padding for structure alignment
Track_Number	21	24	4	int	Track Number (when applicable)
Orbit_Number	25	28	4	int	Orbit number (when applicable)
Reserved	29	32	4	int	Reserved
Number of Frames	33	36	4	int	Number of standard frames[1] (when applicable)

First Frame	37	40	4	int	Num of first standard frame[2] (when applicable)
Reserved	41	56	16	4*int	Reserved
Acquisition_Date	57	62	6	3*short	Acquisition Date in Y M D
Acquisition_Day	63	64	2	short	Acquisition Day of the year
Acquisition_Start	65	72	8	4*short	Start of acquisition in Hours, Min, Sec, Millisec
Acquisition_End	73	80	8	4*short	End of acquisition in Hours, Min, Sec, Millisec
Transcription_Date	81	86	6	3*short	Transcription Date in D M Y (WARNING: Year could be expressed in some tapes as years from 1900)
Reserved	87	98	12	short	Reserved
Filler	99	100	2	char	Padding for structure alignment
Reserved	101	140	40	10*int	Reserved
Physical_Address_1	141	144	4	long	User Header file number[4]
Physical_Address_2	145	148	4	long	Pass Id. Header file number[5]
Reserved	149	845	697	char	Reserved
Copy Date	846	848	3	3*u_char	Date when this record has been generated as a copy from another tape in Day, Month, Year.
Copy Source Tape Number	849	852	4	long	Number of the source tape from where this record was generated
Copy Flag	853	853	1	char	1 = this is a copied record 0 = this is the original transcribed record
Copy Source Media Type	854	854	1	char	4 = DLT 2 = SONY ID1
Reserved	855	856	2	char	Reserved

3. FORMAT CODES

3.1 SATELLITE CODES

LANDSAT	1
MOS	2
J-ERS	3
SPOT	4
ERS	5
IRS-C	6
RADARSAT	7
NOAA	8
RESERVED	9
RESERVED	10
HELIOS	11
SHUTTLE	12
EOSAM	13
EOSPM	14

3.2 INSTRUMENT TYPE CODES

LANDSAT MSS	1
LANDSAT TM	2
LANDSAT ETM	3
LANDSAT RBV	4
MOS MESSR	5
J-ERS VNIR	6
J-ERS SWIR	7
Not Used	8
Not Used	9
ERS AMI SAR	10
ERS ATSR	11
SPOT HRV	12
J-ERS SAR	13
NOAA AVHRR	14
SPOT HRVIR	15
SHUTTLE XSAR	16
MODIS	17

3.3 SOURCE TYPE CODES

AMPEX 14 tracks	1
Shlumberger 14 tracks	2
Shlumberger 42 tracks	3
Penny & Giles	4
Honeywell HD-96	5

AMPEX DCRSi	6
CREO Optical Tape	7
Direct Ingestion	8
SONY DIR 1000 (R)	9

3.4 FORMAT SYNCHRONISER TYPE CODES

MCS ERS FS	1
SPACETEC ERS HR FS	2
IAI ERS HR FS	3
LABEN ERS HR FS	4
SPACETEC/ACS ERS FS	5
ENERTEC MSS FS	6
ENERTEC TM FS	7
ACS SW FORMAT SYNCH	8
ACS XSAR FORMAT SW SYNCH	14

3.5 STATION NAMES CODES

Adelaide	67
Agrhymet	97
AliceSpring	10
Aspendale	68
Atlanta	105
Aussaguel	20
Bangkok	6
BantonRouge	52
Bedford	49
Beijing	74
Berlin	30
Berne	31
Bishkek	112
Bremenhaven	29
Budapest	28
Cairo	96
Casey	92
Cashoiera	64
Copenhagen	32
Cordoba	103
Cotopaxi	8
Cuiaba	23
Da-Xing	76
DeBilt	33
Dhaka	75
Downsview	50
Dundee	34
Edmonton	51
Fairbanks	7

Farnborough	24
Frascati	35
Fucino	1
Funceme	65
Gatineau	9
GilmoreCreek	54
Hamburg	36
HarteBeesHoek	19
Hatoyama	16
Helsinki	37
Hobart	22
Honolulu	69
Hyderabad	11
Islamabad	21
Itu	115
Keelung	77
KhantyMansiysk	116
Kiruna	2
Kiyose	78
Kitab	113
Kourou	111
Krakow	38
KualaLumpur	91
Kumamoto	12
Lannion	39
Lapan	79
LaReunion	99
Lasham	40
Libreville	106
Madrid	41
Malindi	109
Manila	80
Maspalomas	3
Matera	104
McMurdo	93
Miami	53
Moscow	110
Nairobi	98
NESDIS	55
Neustrelitz	108
NewDelhi	81
Norman	102
Norrkoping	42
Oberpfaff	27
Offenbach	43
OHiggins	15
Oslo	44
Palmer	94
PariPari	18
Perth	70
Prague	45
Pretoria	25

PrinceAlbert	13
RedwoodCity	57
Riyadh	82
Rome	46
RRSC_Nairobi	101
Santiago	66
Scanzano	26
Scipps Inst	58
Selangor	84
Sendai	87
Seoul	85
SeoulUniv	86
Singapore	83
SiouxFalls	59
Spitzenbergen	47
StennisSpace	60
Sydney	71
Syowa	17
Taipei	90
Tel_Aviv	5
TerranovaBay	95
TokaiUniv	88
TokyoUniv	89
Townsville	72
Traben-Trar	48
Tromso	4
Tunis	100
UlanBator	114
UnivOfAlaska	61
UnivOfRhodeIsl	63
UnivOfTexas	62
WallopsIsl	56
Wellington	73
WestFreugh	14

4. DLT FORMAT ON DISK

This format is fully equivalent to the DLT format and can be managed by the same software that is able to manage the DLT format. The passage that is archived on DLT in a Transcription Area as described in section 2.3, is stored on disk in a directory. The directory naming convention is the following:

Field	Chars	Values
Format name	5	WILMA (optional field, could be replaced by an upper level directory identifying the archive format)
Separator	1	_ (underscore)
Satellite	3->5	Ers, Lands, Jers, Spot, EOSAM, EOSPM
Mission	1	1 to 9
Separator	1	_ (underscore)
Sensor	2->5	SAR, MSS, TM, ETM, VNIR, HRV, HRVI, MODIS
Separator	1	_ (underscore)
Fixed character	1	T
Track or Orbit	6	000000 -> 999999
Separator	1	_ (underscore)
Fixed character	1	S
Station Code	1->3	One value from table in section 3.5
Separator	1	_ (underscore)
Year	4	Year of acquisition
Month	2	01 -> 12
Day	2	01 -> 31
Separator	1	_ (underscore)
Pass Start Hour	2	00 -> 23
Minutes	2	00 -> 59
Seconds	2	00 -> 59

Here is an example of the directory name for a Landsat 7 ETM pass, track 187, acquired at Matera station the 28/02/2004 and starting at 08:10:15:

WILMA_Lands7_ETM_T187_S104_20020228_081015

The files described in section 2.3 are all inside the storage directory and have a fixed naming convention:

File	Name on disk
Pass Identification Header	DTPassId.dat
Sensor Acquired Data	DTVideoData.dat
User Header	DTUserHeader.dat
Segment Descriptor	DTSegment.dat
Block Address Descriptor	DTBlock.dat
State Vector File	DTOrbitFile.dat
Orbit Data	DTFtdcf
Payload Correction	DTTelemetry.dat
Statistics File	DTStatisticFile.dat