Standard Archive Format for Europe

ENVISAT Specialisation for Level 0 products

Reference	PDGS-SAFE-GMV-ENVISAT-L0	issue 1	revision 0
Author(s)	GMV	Date: 17/12/2015	
Reviewed by	European Space Agency (ESA)	date	
Approved by		date	

ESRIN
Via Galileo Galilei - Casella Postale 64 - 00044 Frascati - Italy
Tel. (39) 06 94J80J - Fax (39) 06 94J80 280

Table of contents

List of Tables

List of Figures

1. Introduction

1.1. Purpose and scope

This document is part of the Standard Archive Format for Europe specialisation for ENVISAT (SAFE Specialisation for ENVISAT). This specialisation consists of the following set of documents:

- the ENVISAT mission specialisation control book, which is the top-level document of the specialisation, containing all the information that is common to all SAFE ENVISAT products.
- one ENVISAT product specialisation control book for ENVISAT Level-0 products.

The current book is the specialisation control book for ENVISAT Level-0 products.

1.2. Book organisation

The specialisation control book for ENVISAT Level-0 products and auxiliary files is organized as follows:

Chapter 1: Introduction	Introductory part of the document.
Chapter 2: Target of preservation	Description of the target of preservation for L0 products.
Chapter 3: Data Structures	Specification of the simple and complex types that are used to represent the structure of the products and auxiliary file types independently of the mission instrument to which they are associated.

1.3. Acronyms and abbreviations

DFDL	Data Format Description Language
GNU	GNU is Not Unix
MDS	Measurement Data Block
MPH	Main Product Header
SPH	Specfic Product Header
W3C	World Wide Web Consortium
XML	eXtensible Mark-up Language

2. Target of preservation

ENVISAT L0 products in native format may be available in tar format (filenames with ".TAR" extension) or alternatively in tar/gzip format (with ".TGZ" extension i.e. GNU-zipped tar file merging). However, the targets of preservation considered in this SAFE specialisation are the contents of those packaged files, i.e. the files which are stored within the tar or tar/gzip files.

Any product in native format must be unpackaged and decompressed before being converted into SAFE and the SAFE Packages will only contain the unpackaged and decompressed files. This is because the representation information schemas that are provided along with this specialisation describe the unpackaged and decompressed files, not the tar/gzip format (there would be limitations in doing this, as explained in the SAFE Core Specifications).

The file structure of all ENVISAT L0 product types is identical and it is described in section 3

3. Data Structures

The information included in this chapter has been generated using the specifications defined by the DFDL schemas that represent the structure of the ENVISAT L0 products.

The representation information is described by means of complex structures that make use of simple types to represent the whole content of a given file type. The following sub-sections provide a detailed description of those types.

The diagrams included in this document provide an overview of the structure of the products by depicting the schemas which provide their representation information.

The following subsections provide a detailed description of the data structures used to represent the information of all ENVISAT L0 products. There are two descriptions depending on the instrument that has been used for the generation of the product (one for MERIS, AATSR, MWR, MIPAS, SCIAMACHY and DORIS and a second one for ASAR, RA-2 and GOMOS L0 products).

The next figure provides a high level overview of the complex structures used to represent the information of the ENVISAT or MERIS, AATSR, MWR, MIPAS, SCIAMACHY and DORIS L0 products:

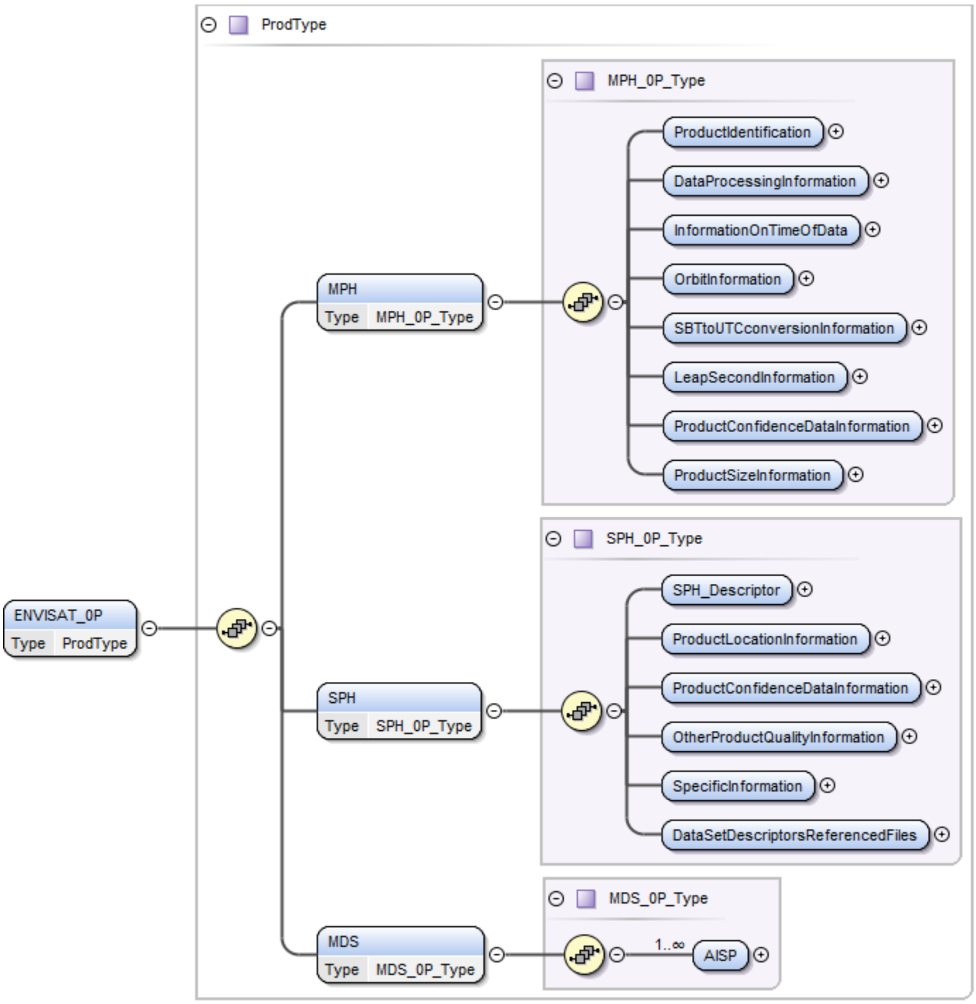

Figure 1: DFDL schema organisation for ENVISAT L0 product files (MERIS, AATSR, MWR, MIPAS, SCIAMACHY and DORIS L0 products)

The next figure provides a high level overview of the complex structures used to represent the information of the ENVISAT ASAR, RA-2 and GOMOS L0 products:

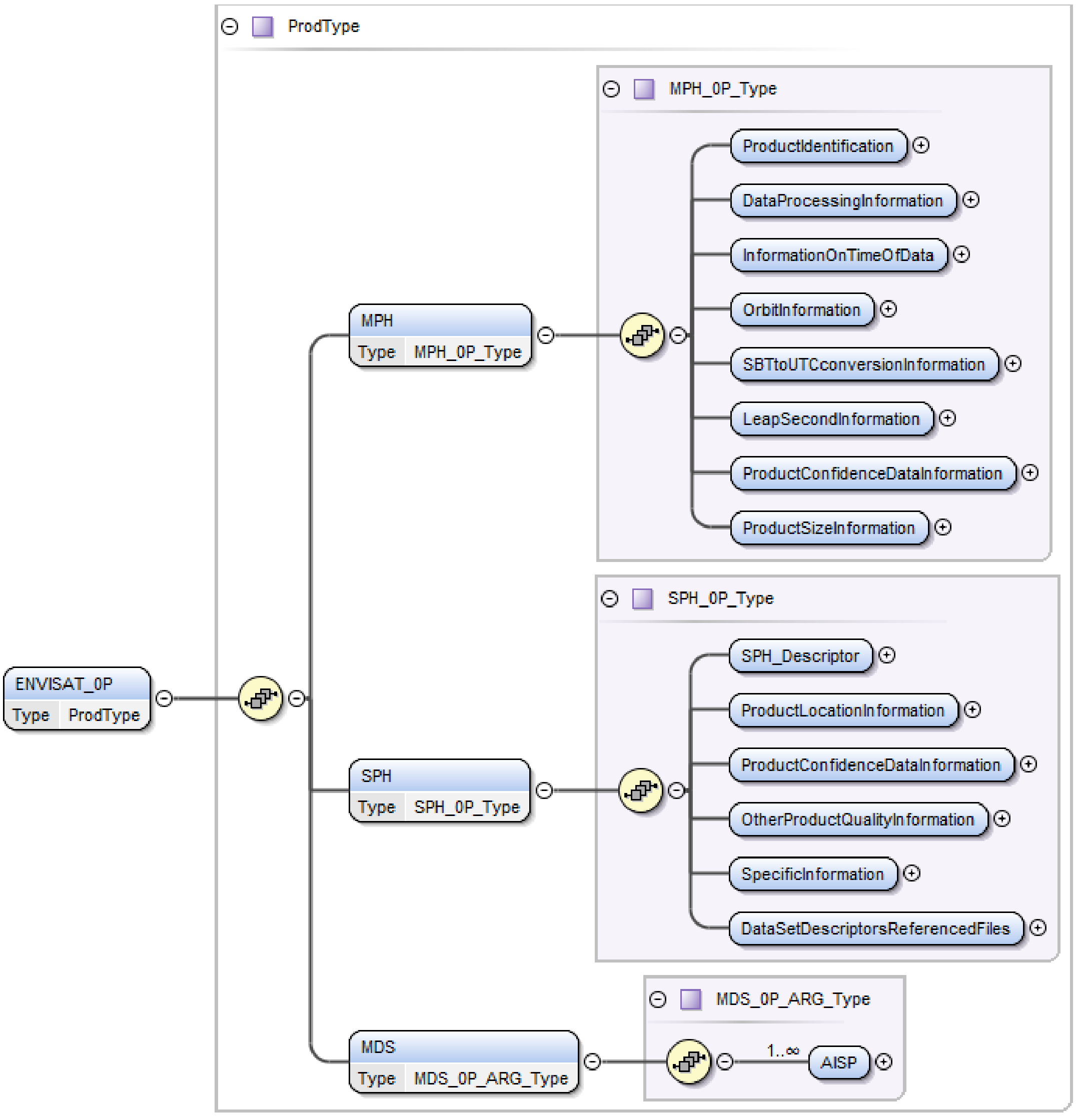

Figure 2: DFDL schema organisation for ENVISAT L0 product files (ASAR, RA-2 and GOMOS)

3.1. Root Element

#	Name/Description	Format
1	**ENVISAT_0P** Product File in ENVISAT Format The Product File structure will follow the one defined for the ENVISAT level-0 products. Each level-0 file is composed by: * Main Product Header (MPH) * Specific Product Header (SPH) * Measurement Data Block The Measurement Data Block contains one ENVISAT Measurement Data Set (MDS) for each file.	ProductType (See 3.1.1. for MERIS, AATSR, MWR, MIPAS, SCIAMACHY and DORIS) (See 3.1.2. for ASAR, RA-2 and GOMOS)

Table 1: ENVISAT_0P Specification

3.1.1. ProdType (for MERIS, AATSR, MWR, MIPAS, SCIAMACHY and DORIS)

#	Name/Description	Format
1	**MPH**	MPH_0P_Type
2	**SPH**	SPH_0P_Type
3	**MDS**	MDS_0P_Type

Table 2: ProdType Specification

3.1.2. ProdType (for ASAR, RA-2 and GOMOS)

#	Name/Description	Format
1	**MPH**	MPH_0P_Type
2	**SPH**	SPH_0P_Type
3	**MDS**	MDS_0P_ARG_Type

Table 3: ProdType Specification

3.2. Basic Types

The following basic types have been defined to be used by other complex types.

3.2.1. uc

Base Type	Length (bytes)	Comments
xs:string	1	Unsigned char (uc)

Table 4: uc Specification

3.2.2. utc

Base Type	Length (bytes)	Comments
xs:dateTime		UTC time (27 bytes) dd-MMM-yyyy hh:mm:ss.uuuuuu Pattern: dd-MMM-yyy HH:mm:ss.SSSSSS

Table 5: utc Specification

3.2.3. int_s04d

Base Type	Length (bytes)	Comments
xs:short	4	Pattern: +000;-000

Table 6: int_s04d Specification

3.2.4. int_s06d

Base Type	Length (bytes)	Comments
xs:int	6	Pattern: +00000;-00000

Table 7: int_s06d Specification

3.2.5. int_06d

Base Type	Length (bytes)	Comments
xs:int	6	Pattern: 000000

Table 8: int_06d Specification

3.2.6. int_s07d

Base Type	Length (bytes)	Comments
xs:int	7	Pattern: +000000;-000000

Table 9: int_s07d Specification

3.2.7. int_07d

Base Type	Length (bytes)	Comments
xs:int	7	Pattern: 0000000

Table 10: int_07d Specification

3.2.8. int_s11d

Base Type	Length (bytes)	Comments
xs:long	11	Pattern: +0000000000;-0000000000

Table 11: int_s11d Specification

3.2.9. int_s21d

Base Type	Length (bytes)	Comments
xs:unsignedLong	21	Pattern: +00000000000000000000;-00000000000000000000

Table 12: int_s21d Specification

3.2.10. real_s08_6f

Base Type	Length (bytes)	Comments
xs:decimal	8	Pattern: +.00000;-.00000

Table 13: real_s08_6f Specification

3.2.11. real_s11_6f

Base Type	Length (bytes)	Comments
xs:decimal	11	Pattern: +0000.000000;-0000.000000

Table 14: real_s11_6f Specification

3.2.12. real_11_6f

Base Type	Length (bytes)	Comments
xs:decimal	11	Pattern: 0000.000000

Table 15: real_11_6f Specification

3.2.13. real_s12_3f

Base Type	Length (bytes)	Comments
xs:decimal	12	Pattern: +0000000.000;-0000000.000

Table 16: real_s12_3f Specification

3.2.14. real_s12_6f

Base Type	Length (bytes)	Comments
xs:decimal	12	Pattern: +0000.000000;-0000.000000

Table 17: real_s12_6f Specification

3.2.15. MJDType

(Modified Julian Day 2000) is the decimal number of day since January 1, 2000 at 00:00 hours. It is represented by 3 long integers (4 bytes each, 12 bytes total)

#	Name/Description	Format
1	**daysElapsed** Number of days elapsed since the 1st of January 2000 at 0:0 hour. It may be negative, and is thus a signed long integer	xs:long 4 bytes
2	**secondsElapsed** Number of seconds elapsed since the beginning of that day	xs:unsignedLong 4 bytes
3	**microsecondsElapsed** Number of microseconds elapsed since the last second	xs:unsignedLong 4 bytes

Table 18: MJDType Specification

3.3. Complex Types

3.3.1.1. MPH_0P_Type

#	Name/Description	Format
1	**ProductIdentification**	ProductIdentificationType
2	**DataProcessingInformation**	DataProcessingInformationType
3	**InformationOnTimeOfData**	InformationOnTimeOfDataType
4	**OrbitInformation**	OrbitInformationType
5	**SBTtoUTCconversionInformation**	SBTtoUTCconversionInformationType
6	**LeapSecondInformation**	LeapSecondInformationType
7	**ProductConfidenceDataInformation**	MPHProductConfidenceDataInformation-Type
8	**ProductSizeInformation**	ProductSizeInformationType

Table 19: MPH_0P_Type Specification

3.3.1.1.1. ProductIdentificationType

Product Identification Information.
This information includes the file name of the product (which describes most of the essential features of the product, such as instrument, mode, and processing level), the consolidation level of the product, and the document ID number of the documentation describing this product.

#	Name/Description	Format
1	**PRODUCT="**	xs:string 9 bytes
	Product File Name It is left justified with trailer blanks Note: the file name shall be without the extension	xs:string 62 bytes
	"	uc 1 bytes
2	**PROC_STAGE=**	xs:string 11 bytes

	Processing stage code:
	N = Near Real Time
	T = Test Product
	V = fully validated (fully consolidated) product
	S = Special product
	Letters between N and V (with the exception of T and S) indicate steps in the consolidation process, with letters closer to V meaning higher levels of consolidation.
	If not used, set to X.
		uc

3	**REF_DOC="**	xs:string 9 bytes
	Reference Document Describing Product AA-BB-CCC-DD-EEEE_V/IØØ (23 characters,	xs:string 23 bytes

#	Name/Description	Format
	including blank space characters) where AA-BB-CCC-DD-EEEE is the ESA standard document no. and V/I is the Version / Issue If not used, set to ØØØØØØØØØØØØØØØØØØØØØØØ	
	"	uc 1 bytes
4	**Spare**	xs:string 40 bytes

Table 20: ProductIdentificationType Specification

3.3.1.1.2. DataProcessingInformationType

Information Regarding Data Acquisition and Processing.
This information identifies where the product was acquired, where it was processed, when it was processed, and what hardware/software performed the processing.

#	Name/Description	Format
1	**ACQUISITION_STATION="**	xs:string 21 bytes
	Acquisition Station ID (up to 3 codes from:) Multiple entries are separated by commas e.g. PDHS-K,DPAC, LRACØØØ. String is left justified with blank space characters used for unused characters. If not used, set to 20 blanks. Possible values: PDHS-K PDHS-E LRAC PDCC FOS-ES PDAS-F UK-PAC D-PAC I-PAC F-PAC S-PAC E-PAC ECMWF	xs:string 20 bytes
	"	uc 1 bytes
2	**PROC_CENTER="**	xs:string 13 bytes

#	Name/Description	Format
	Processing Center ID code Processing Center ID which generated current product (Note 3) (1 site code according to list in Field 5 above, left justified). If not used, set to 6 blanks. Possible values: PDHS-K PDHS-E LRAC PDCC FOS-ES PDAS-F UK-PAC D-PAC I-PAC F-PAC S-PAC E-PAC ECMWF	xs:string 6 bytes
	"	uc 1 bytes
3	**PROC_TIME="**	xs:string 11 bytes
	Processing Time (Product Generation Time in UTC Time format) If not used, set to blanks.	xs:string 27 bytes
	"	uc 1 bytes
4	**SOFTWARE_VER="**	xs:string 14 bytes
	Software Version number of processing software Format: Name of processor (up to 10 characters)/ version number (4 characters) -- left justified (any blanks added at end). If not used set to 14 blanks e.g. "MIPAS/2.31 "	xs:string 14 bytes
	"	uc 1 bytes
5	**Spare**	xs:string 40 bytes

Table 21: DataProcessingInformationType Specification

3.3.1.1.3. InformationOnTimeOfDataType

Information on Time of Data. Included in these fields are the UTC start and stop time of data sensing.

#	Name/Description	Format
1	**SENSING_START="**	xs:string 15 bytes
	UTC start time of data sensing (first measurement in first data record). UTC Time format. If not used set to 27 blanks	xs:string 27 bytes
	"	uc 1 bytes
2	**SENSING_STOP="**	xs:string 14 bytes
	UTC stop time of data sensing (last measurements last data record). UTC Time format. If not used set to 27 blanks	xs:string 27 bytes
	"	uc 1 bytes
3	**Spare**	xs:string 40 bytes

Table 22: InformationOnTimeOfDataType Specification

3.3.1.1.4. OrbitInformationType

Information on ENVISAT Orbit and Position.
These fields contain orbit positioning data which allow one to determine the exact position of the satellite at the time of sensing.

#	Name/Description	Format
1	**PHASE=**	xs:string 6 bytes
	Phase Code: phase letter (A, B, ...) If not used set to X	uc
2	**CYCLE=**	xs:string 6 bytes
	Cycle number. If not used set to +000	xs:string 4 bytes
3	**REL_ORBIT=**	xs:string 10 bytes
	Relative Orbit Number at sensing start time. If not used set to +00000	xs:integer 6 bytes
4	**ABS_ORBIT=**	xs:string 10 bytes
	Absolute Orbit Number at sensing start time. If not used set to +00000	int_s06d
5	**STATE_VECTOR_TIME="**	xs:string 19 bytes
	UTC state vector time It is filled properly in case of usage of FOS Predicted Orbit information otherwise it shall be set	xs:string 27 bytes

#	Name/Description	Format
	to 27 blanks.	
	"	uc 1 bytes
6	**DELTA_UT1=**	xs:string 10 bytes
	Universal Time Correction <s> DUT1 = UT1 – UTC IF not used set to +.000000<s>	xs:string 11 bytes
7	**X_POSITION=**	xs:string 11 bytes
	X position in Earth Fixed Reference: value<m> It is filled properly in case of usage of FOS Predicted Orbit information otherwise it shall be set to +0000000.000<m>	xs:string 15 bytes
8	**Y_POSITION=**	xs:string 11 bytes
	Y position in Earth Fixed Reference: value<m> It is filled properly in case of usage of FOS Predicted Orbit information otherwise it shall be set to +0000000.000<m>	xs:string 15 bytes
9	**Z_POSITION=**	xs:string 11 bytes
	Z position in Earth Fixed Reference: value<m> It is filled properly in case of usage of FOS Predicted Orbit information otherwise it shall be set to +0000000.000<m>	xs:string 15 bytes
10	**X_VELOCITY=**	xs:string 11 bytes
	X velocity in Earth Fixed Reference: value<m/s> It is filled properly in case of usage of FOS Predicted Orbit information otherwise it shall be set to +0000.000000<m/s>	xs:string 17 bytes
11	**Y_VELOCITY=**	xs:string 11 bytes
	Y velocity in Earth Fixed Reference: value<m/s> It is filled properly in case of usage of FOS Predicted Orbit information otherwise it shall be set to +0000.000000<m/s>	xs:string 17 bytes
12	**Z_VELOCITY=**	xs:string 11 bytes
	Z velocity in Earth Fixed Reference: value<m/s> It is filled properly in case of usage of FOS Predicted Orbit information otherwise it shall be set to +0000.000000<m/s>	xs:string 17 bytes
13	**VECTOR_SOURCE="**	xs:string 15 bytes
	Source Record of Orbit State Vector	xs:string 2 bytes
	"	uc

#	Name/Description	Format
		1 bytes
14	**Spare**	xs:string 40 bytes

Table 23: OrbitInformationType Specification

3.3.1.1.5. SBTtoUTCconversionInformationType

SBT to UTC Conversion Information.
This data allows for precise conversion from Satellite Binary Time (as stored in Instrument Source Packets) to the conventional UTC time system.

#	Name/Description	Format
1	**UTC_SBT_TIME="**	xs:string 14 bytes
	UTC time corresponding to SBT below (currently defined to be given at the time of the ascending node state vector). If not used, set to 27 blanks	xs:string 27 bytes
	"	uc 1 bytes
2	**SAT_BINARY_TIME=**	xs:string 16 bytes
	Satellite Binary Time 32bit integer time of satellite clock. If not used, set to +0000000000. (This value is unsigned, i.e., to be interpreted ≥0)	xs:string 11 bytes
3	**CLOCK_STEP=**	xs:string 11 bytes
	Clock Step Size: Clock step in picoseconds. If not used, set to +0000000000. (This value is unsigned, i.e., to be interpreted ≥0)	xs:string 15 bytes
4	**Spare**	xs:string 32 bytes

Table 24: SBTtoUTCconversionInformationType Specification

3.3.1.1.6. LeapSecondInformationType

#	Name/Description	Format
1	**LEAP_UTC="**	xs:string 10 bytes
	UTC Time of the occurrence of the leap second (dd-MMM-yyyy hh:mm:ss.uuuuuu) If a leap second occurred in the product window the field is set by a devoted function in the CFI EXPLORER_LIB library (see [EXPL_LIB-SUM] for details), otherwise it is set to 27 blanks. It corresponds to the time after the leap second oc-	xs:string 27 bytes

#	Name/Description	Format
	currence (i.e.: midnight of the day after the leap second).	
	"	uc 1 bytes
2	**LEAP_SIGN=**	xs:string 10 bytes
	Leap second sign. If a leap second occurred in the product window the field is set to the expected value by a devoted function in the CFI EXPLORER_LIB library (see [EXPL_LIB-SUM] for details), otherwise it is set to +000.	int_s04d
3	**LEAP_ERR=**	xs:string 9 bytes
	Leap second error flag. Leap second error if leap second occurs within processing segment = 1, otherwise = 0. If not used, set to 0.	uc
4	**Spare**	xs:string 40 bytes

Table 25: LeapSecondInformationType Specification

3.3.1.1.7. MPHProductConfidenceDataInformationType

Product Confidence Data.
Product Confidence Data in the MPH is designed to very simply provide the user with an assessment of the overall product quality by reporting if errors have occurred during the processing.
To obtain a detailed description of the errors which occurred the user refers to the SPH or the detailed PCD structures of the product.

#	Name/Description	Format
1	**PRODUCT_ERR=**	xs:string 12 bytes
	Product Error Flag set to 1 if errors have been reported in the product. The flag is set to 1 if at least one of the following field of the SPH have reported errors: NUM_MISS_ISPS NUM_ERR_ISPS NUM_DISCARDED_ISPS	uc

Table 26: MPHProductConfidenceDataInformationType Specification

3.3.1.1.8. ProductSizeInformationType

Product Size Information.
These fields identify the size of various structures within the product so that they may be accurately interpreted.

#	Name/Description	Format
1	**TOT_SIZE=**	xs:string 9 bytes
	Total size of the product: value<bytes>	xs:string 28 bytes
2	**SPH_SIZE=**	xs:string 9 bytes
	Length of the SPH: value>bytes<	xs:string 18 bytes
3	**NUM_DSD=**	xs:string 8 bytes
	Number of Data Set Descriptors,including spares and all other types of DSDs	int_s11d
4	**DSD_SIZE=**	xs:string 9 bytes
	Length of the DSD: value>bytes<	xs:string 18 bytes
5	**NUM_DATA_SETS=**	xs:string 14 bytes
	Number of attached Data Sets (note that not all the DSDs have a DS attached)	int_s11d
6	**Spare**	xs:string 40 bytes

Table 27: ProductSizeInformationType Specification

3.3.1.2. SPH_0P_Type

#	Name/Description	Format
1	**SPH_DESCRIPTOR="**	xs:string 16 bytes
	Format: <Product ID>ØSPECIFICØHEADER where the product ID contains 10 characters. If needed, blanks are located to the right of the last character in the text string. e.g. MER_RR__0PØSPECIFICØHEADERØØ	xs:string 28 bytes
	"	uc 1 bytes
2	**ProductLocationInformation**	ProductLocationInformationType
3	**ProductConfidenceDataInformation**	ProductConfidenceDataInformationType
4	**OtherProductQualityInformation**	OtherProductQualityInformationType
5	**ASARSpecificInformation**	SpecificInformationType
6	**DataSetDescriptorsReferencedFiles**	DataSetDescriptorsReferencedFilesType

Table 28: SPH_0P_Type Specification

3.3.1.2.1. ProductLocationInformationType

#	Name/Description	Format
1	**START_LAT=**	xs:string

#	Name/Description	Format
		10 bytes
	WGS84 latitude of first satellite nadir point at the Sensing Start time of the MPH (positive north) (units: <10-6degN>)	xs:string 21 bytes
2	**START_LONG=**	xs:string 11 bytes
	WGS84 longitude of first satellite nadir point at the Sensing Start time of the MPH (positive East, 0 = Greenwich). (units: <10-6degN>)	xs:string 21 bytes
3	**STOP_LAT=**	xs:string 9 bytes
	WGS84 latitude of first satellite nadir point at the Sensing Stop time of the MPH (positive north) (units: <10-6degN>)	xs:string 21 bytes
4	**STOP_LONG=**	xs:string 10 bytes
	WGS84 longitude of first satellite nadir point at the Sensing Stop time of the MPH (positive East, 0 = Greenwich). (units: <10-6degN>)	xs:string 21 bytes
5	**SAT_TRACK=**	xs:string 10 bytes
	Sub-satellite track heading at the Sensing Start time in the MPH. (units: <deg>)	xs:string 20 bytes
6	**Spare**	xs:string 50 bytes

Table 29: ProductLocationInformationType Specification

3.3.1.2.2. ProductConfidenceDataInformationType

#	Name/Description	Format
1	**ISP_ERRORS_SIGNIFICANT=**	xs:string 23 bytes
	1 or 0. 1 if number of ISPs with CRC errors exceeds threshold For ERS, always set to a default value taken from PRM_Level0SphDefaults.dat parameter file.	uc
2	**MISSING_ISPS_SIGNIFICANT=**	xs:string 25 bytes
	1 or 0. 1 if number of missing ISPs exceeds threshold. For ERS, always set to a default value taken from PRM_Level0SphDefaults.dat parameter file.	uc
3	**ISP_DISCARDED_SIGNIFICANT=**	xs:string 26 bytes
	1 or 0. 1 if number of ISPs discarded by the PF-HS exceeds threshold.	uc

#	Name/Description	Format
	For ERS, always set to a default value taken from PRM_Level0SphDefaults.dat parameter file.	
4	**RS_SIGNIFICANT=**	xs:string 15 bytes
	1 or 0. 1 if number of ISPs with Reed Solomon corrections exceeds threshold. For ERS, always set to a default value taken from PRM_Level0SphDefaults.dat parameter file.	uc
5	**Spare**	xs:string 50 bytes

Table 30: ProductConfidenceDataInformationType Specification

3.3.1.2.3. OtherProductQualityInformationType

#	Name/Description	Format
1	**NUM_ERROR_ISPS=**	xs:string 15 bytes
	Number of ISPs containing CRC errors.	xs:string 11 bytes
2	**ERROR_ISPS_THRESH=**	xs:string 18 bytes
	Threshold at which number of ISPs containing CRC errors is considered significant. For ERS, this is set to the value in the Level 0 Processor Configuration File.	xs:string 18 bytes
3	**NUM_MISSING_ISPS=**	xs:string 17 bytes
	Number of missing ISPs	xs:string 11 bytes
4	**MISSING_ISPS_THRESH=**	xs:string 20 bytes
	Number of ISPs discarded	xs:string 18 bytes
5	**NUM_DISCARDED_ISPS=**	xs:string 19 bytes
	Number of ISPs discarded by PF-HS. For ERS, always set to a default value taken from PRM_Level0SphDefaults.dat parameter file.	xs:string 11 bytes
6	**DISCARDED_ISPS_THRESH=**	xs:string 22 bytes
	Threshold at which number of ISPs discarded by PF-HS is considered significant. For ERS, this is set to the value in the Level 0 Processor Configuration File.	xs:string 18 bytes
7	**NUM_RS_ISPS=**	xs:string 12 bytes
	Number of ISPs with Reed Solomon corrections.	xs:string

#	Name/Description	Format
	For ERS, always set to a default value taken from PRM_Level0SphDefaults.dat parameter file.	11 bytes
8	**RS_THRESH=**	xs:string 10 bytes
	Number of ISPs with Reed Solomon corrections. For ERS, always set to a default value taken from PRM_Level0SphDefaults.dat parameter file.	xs:string 18 bytes
9	**Spare**	xs:string 100 bytes

Table 31: OtherProductQualityInformationType Specification

3.3.1.2.4. SpecificInformationType

#	Name/Description	Format
1	**TX_RX_POLAR="**	xs:string 13 bytes
	Polarization (used for ASAR only) HV/HV, H/HVØ, V/VHØ, H/HØØ, H/VØØ, or V/VØØ, or V/HØØ The letter(s) to the left of the '/' indicates the transmitter polarization. The letter(s) to the right of the '/' indicates the receiver polarization. ØØØØØ for non-ASAR products.	xs:string 5 bytes
	"	uc 1 bytes
2	**SWATH="**	xs:string 7 bytes
	Swath Number (used for ASAR only) codes: IS1, IS2, IS3, IS4, IS5, IS6, IS7,WSØ -- WS is used for WS mode and GM mode. For ASA_EC__OP and ASA_MS__OP, the field is set to EC0 and MC0, respectively. ØØØ for non-ASAR products.	xs:string 3 bytes
	"	uc 1 bytes
3	**Spare**	xs:string 41 bytes

Table 32: ASARSpecificInformationType Specification

3.3.1.2.5. DataSetDescriptorsReferencedFilesType

#	Name/Description	Format
1	**DSD_MDS** DSD (M) for MDS	DSDType
2	**DSD_CONF_FILE** DSD (R) pointing to Configuration file	DSDType

#	Name/Description	Format
3	**DSD_OSV_FILE** DSD (R) pointing to the Orbit State Vector file used	DSDType
4	**Spare**	xs:string 279 bytes

Table 33: DataSetDescriptorsReferencedFilesType Specification

3.3.1.2.6. DSDType

#	Name/Description	Format
1	**DS_NAME="**	xs:string 9 bytes
	Name describing the Data Set Possible values: ASAR_SOURCE_PACKETS AATSR_SOURCE_PACKETS DORIS_SOURCE_PACKETS GOMOS_SOURCE_PACKETS MERIS_SOURCE_PACKETS MIPAS_SOURCE_PACKETS MWR_SOURCE_PACKETS RA2_SOURCE_PACKETS SCIAMACHY_SOURCE_PACKETS HOUSEKEEPING_PACKETS	xs:string 28 bytes
	"	uc 1 bytes
2	**DS_TYPE=**	xs:string 8 bytes
	Type of the Data Set. Possible values: M R G	uc
3	**FILENAME="**	xs:string 10 bytes
	Name of the Reference File. Not used. Set to 62 blanks	xs:string 62 bytes
	"	uc 1 bytes
4	**DS_OFFSET=**	xs:string 10 bytes
	Offset in bytes (MPH+SPH including DSD): value<bytes>	xs:string 28 bytes
5	**DS_SIZE=**	xs:string 8 bytes
	Size in bytes of the Attached Data Set: value<bytes> Used if DS_TYPE is set to M If not used set to	xs:string 28 bytes

#	Name/Description	Format
	00000000000000000000<bytes>	
6	**NUM_DSR=**	xs:string 8 bytes
	Number of Data Set Records in the Attached Data Set	int_s11d
7	**DSR_SIZE=**	xs:string 9 bytes
	Size of the Data Set Record in the Attached Data Set: value<bytes> If variable set to –0000000001<bytes> If not used set to +0000000000<bytes>	xs:string 18 bytes
8	**Spare**	xs:string 32 bytes

Table 34: DSDType Specification

3.3.1.3. MDS_0P_Type

MDS struct for all ENVISAT instruments except for ASAR, RA-2 and GOMOS (MDS_0P_ARG_Type should be used instead)

#	Name/Description	Format
1	**AISP**	AISPType Min Occurs: 1 Max Occurs: unbounded

Table 35: MDS_0P_Type Specification

3.3.1.3.1. AISPType

#	Name/Description	Format
1	**Annotations**	AnnotationsType
2	**ISP**	InstrumentSourcePacketType

Table 36: AISPType Specification

3.3.1.3.2. InstrumentSourcePacketType

The Source Packet must, in addition to the source data, carry a minimum of information needed by the ground data capture system for the acquisition, storage and distribution of the source data to the end user. Thus, the Source Packet format consists of two major fields:

- The Packet Header, of fixed length, which provides the standardised control information required during the end-to-end transport process from the source on board the spacecraft to the end-user data processing equipment on the ground.

- The Packet Data Field, of variable length, which contains the source data.

The standardised control information that the Source Packet Header must provide is the following:
- Identification of the source and its application process: for data distribution, storage and retriev-

al.
- Sequence numbering for a given source and its application process: for sequence tracking and accounting.
- Packet Data Field length: information used throughout the transport process.

The only other constraint placed on the data source is that the length of the Packet Data Field must not exceed 216 (65536) octets.

#	Name/Description	Format
1	**Packet_Header**	Packet_HeaderType
2	**Packet_Data_Field**	Packet_Data_FieldType

Table 37: InstrumentSourcePacketType Specification

3.3.1.3.3. Packet_Data_FieldType

#	Name/Description	Format
1	**Data_field_header** The Data Field Header is an optional subdivision of the Packet Data Field. The purpose of the Data Field Header is to provide a standard means for inserting within the first octets of a Source Packet Data Field any ancillary data (time, additional packet type identification, internal data field format identification, etc.) which may be necessary to permit the interpretation of the source data contained within the packet by common data-processing facilities. The presence or absence of a Data Field Header must be signalled by the Data Field Header Flag in the Packet Header. The length of the Data Field Header shall be a multiple (integer) of octets. Note that the Data Field Header for all ENVISAT instruments has been standardized such that the first 3 fields are: * Data Field Header Length (16 bits), * Instrument Mode Definition (16 bits), * ICU OBT code (32 bits).	Data_field_headerType
2	**Source_data**	Source_dataType

Table 38: Packet_Data_FieldType Specification

3.3.1.3.4. Data_field_headerType

#	Name/Description	Format
1	**Data_field_header_length**	xs:hexBinary 2 bytes
2	**Instrument_mode_definition** Instrument Mode Words:	xs:hexBinary 2 bytes

#	Name/Description	Format
	Mode Identifier Code 54 Image Mode 5B Wide Swath 98 Wave Mode AB Global Monitoring 67 AP Co-Polar Mode 68 AP Cross-Polar H A4 AP Cross-Polar V	
3	**ICU_OBT_code**	xs:hexBinary 4 bytes

Table 39: Data_field_headerType Specification

3.3.1.4. MDS_0P_ARG_Type

MDS struct for ASAR, RA-2 and GOMOS

#	Name/Description	Format
1	**AISP**	AISP_ARG_Type Min Occurs: 1 Max Occurs: unbounded

Table 40: MDS_0P_ARG_Type Specification

3.3.1.4.1. AISP_ARG_Type

#	Name/Description	Format
1	**Annotations**	AnnotationsType
2	**ISP**	InstrumentSourcePacket_ARG_Type

Table 41: AISP_ARG_Type Specification

3.3.1.4.2. InstrumentSourcePacket_ARG_Type

The Source Packet must, in addition to the source data, carry a minimum of information needed by the ground data capture system for the acquisition, storage and distribution of the source data to the end user. Thus, the Source Packet format consists of two major fields:

- The Packet Header, of fixed length, which provides the standardised control information required during the end-to-end transport process from the source on board the spacecraft to the end-user data processing equipment on the ground.
- The Packet Data Field, of variable length, which contains the source data.

The standardised control information that the Source Packet Header must provide is the following:

- Identification of the source and its application process: for data distribution, storage and retrieval.
- Sequence numbering for a given source and its application process: for sequence tracking and accounting.
- Packet Data Field length: information used throughout the transport process.

The only other constraint placed on the data source is that the length of the Packet Data Field must not exceed 216 (65536) octets.

#	Name/Description	Format
1	**Packet_Header**	Packet_HeaderType
2	**Packet_Data_Field**	Packet_Data_Field_ARG_Type

Table 42: InstrumentSourcePacket_ARG_Type Specification

3.3.1.4.3. Packet_Data_Field_ARG_Type

#	Name/Description	Format
1	**Data_field_header** The Data Field Header is an optional subdivision of the Packet Data Field. The purpose of the Data Field Header is to provide a standard means for inserting within the first octets of a Source Packet Data Field any ancillary data (time, additional packet type identification, internal data field format identification, etc.) which may be necessary to permit the interpretation of the source data contained within the packet by common data-processing facilities. The presence or absence of a Data Field Header must be signalled by the Data Field Header Flag in the Packet Header. The length of the Data Field Header shall be a multiple (integer) of octets. Note that the Data Field Header for all ENVISAT instruments has been standardized such that the first 3 fields are: * Data Field Header Length (16 bits), * Instrument Mode Definition (16 bits), * ICU OBT code (48 bit codes, only for ASAR, RA-2, and GOMOS).	Data_field_header_ARG_Type
2	**Source_data**	Source_dataType

Table 43: Packet_Data_Field_ARG_Type Specification

3.3.1.4.4. Data_field_header_ARG_Type

#	Name/Description	Format
1	**Data_field_header_length**	xs:hexBinary 2 bytes
2	**Instrument_mode_definition** Instrument Mode Words: Mode Identifier Code 54 Image Mode 5B Wide Swath	xs:hexBinary 2 bytes

#	Name/Description	Format
	98 Wave Mode AB Global Monitoring 67 AP Co-Polar Mode 68 AP Cross-Polar H A4 AP Cross-Polar V	
3	**ICU_OBT_code_ARG**	xs:hexBinary 6 bytes

Table 44: Data_field_header_ARG_Type Specification

3.3.1.5. AnnotationsType

#	Name/Description	Format
1	**ISP_Sensing_Time**	MJDType
2	**FEP_Annotations** FEP Quality Data and reception time stamp (Ground Segment Reference Time)	FEP_AnnotationsType

Table 45: AnnotationsType Specification

3.3.1.5.1. FEP_AnnotationsType

#	Name/Description	Format
1	**MJ2000** Ground Station Reference Time of reception.	MJDType
2	**Length_ISP** Length of ISP = (length of source packet excluding 6 byte header) - 1. This field uses the same definition as the Packet Length field of the Packet Header.	xs:unsignedShort 2 bytes
3	**VCDUs_CRC** Number of VCDUs in the ISP which contain a CRC error. As identified by a failed Cyclic Redundancy Code check.	xs:unsignedShort 2 bytes
4	**VCDUs_RS** Number of VCDUs in the ISP for which a Reed-Solomon error correction was performed	xs:unsignedShort 2 bytes
5	**Spare**	xs:hexBinary 2 bytes

Table 46: FEP_AnnotationsType Specification

3.3.1.6. Packet_HeaderType

#	Name/Description	Format
1	**Packet_identification**	xs:unsignedShort 2 bytes

#	Name/Description	Format
2	**Packet_seq_control**	xs:unsignedShort 2 bytes
3	**Packet_length** Length of ISP = (length of source packet excluding 6 byte header)-1 bytes	xs:unsignedShort 2 bytes

Table 47: Packet_HeaderType Specification

3.3.1.7. Packet_identificationType

Packet Identification is a 16-bit field divided into four subfields namely Version Number (3 bits), Reserved Bit (1 bit), Data Field Header Flag (1 bit) and Application Process Identifier (11 bits).

#	Name/Description	Format
1	**versionNumber** The Version Number is a 3-bit field occupying the three most significant bits of a packet structure. The Version Numbers are defined by the CCSDS. IN THIS STANDARD, ONLY ONE VERSION NUMBER (VERSION 2) IS PERMITTED, and this specifies the packet formats described in this Section.	xs:unsignedShort 3bits
2	**type** Packets may be identified to be either telemetry type (Bit 3=0) or telecommand type (Bit 3 = 1). All telemetry Source Packets shall have this bit set to ‘0’.	xs:unsignedShort 1bits
3	**Data_field_headerFlag** The Data Field Header Flag indicates the presence (Bit 4= 1) or absence (Bit 4= 0) of a Data Field Header within the Packet Data Field.	xs:unsignedShort 1bits
4	**APID** The Application Process Identifier is an 11 -bit field uniquely identifying both the physical source (instrument or subsystem unit) and the particular application process within this physical source which created the Source Packet. A physical source may ‘own’ more than one application process. Any Identifier is unique on board a given spacecraft, regardless of the number of Virtual Channels used. The Application Process Identifiers are tailored to the mission needs, in general, and to the overall data handling system requirements, in particular. They are ultimately assigned by the Mission Control authority. Each Application Process Identifier is logically as-	xs:unsignedShort 11bits

#	Name/Description	Format
	sociated with the Source Sequence Count subfield of the Packet Sequence Control field. This is to allow the ground telemetry acquisition systems to control the continuity of packet delivery for each Application Process ID.	

Table 48: Packet_identificationType Specification

3.3.1.8. Packet_seq_controlType

#	Name/Description	Format
1	**sequenceFlags** These Flags occupy the two most-significant bits of the 16-bit field. In the Source Packet, these Flags shall always be set to ‘all ones’.	xs:unsignedShort 2bits
2	**sequenceCount** This 14-bit field contains a straight sequential count (modulo 16 384) of each packet generated by each unique source application process (as specified by the Application Process ID) on the spacecraft. The field will allow the ground telemetry acquisition systems to control the continuity of packet delivery for each Application Process ID. During the continuous operation of a source application process, it is not permissible for the source to ‘short cycle’ the sequence counter by resetting before the full counter accumulation has been reached; however, if the operation of a source is interrupted (e.g. through the power supply’s being switched off), the source may start a new sequence count when its operation is resumed. The source application process responsible for generating the Idle Source Packets (Application Process ID ‘all ones’) is not required to maintain a Source Sequence Count.	xs:unsignedShort 14bits

Table 49: Packet_seq_controlType Specification

3.3.1.9. Source_dataType

#	Name/Description	Format
1	**Application_data** The Application Data field is a subdivision of the Packet Data Field. It is the user data in the form of a sequence of octets. Application_data_length = Packet_length -	xs:hexBinary {../../../Packet_Header/Packet_length - 10 + 1} bytes

#	Name/Description	Format
	Data_field_header (length) + 1	

Table 50: Source_dataType Specification