Standard Archive Format for Europe

GOCE Specialisation for Level 0 processing

Reference	PDGS-SAFE-GMV-GOCE-L0	issue 1	revision 1d
Author(s)	GMV	date: 28/08/2014	
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 GOCE (SAFE Specialisation for GOCE). This specialisation consists of the following set of documents:

- the GOCE mission specialisation control book, which is the top-level document of the specialisation, containing all the information that is common to all SAFE GOCE products and auxiliary files.
- three GOCE product specialisation control books organized by product level, one for GOCE Level-0 products and auxiliary files, one for Level-1 auxiliary files and one for GOCE Level-2 products.

The current book is the specialisation control book for GOCE Level-0 products and auxiliary files.

1.2. Book organisation

The specialisation control book for GOCE 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 and auxiliary files.
Chapter 3: Data Structures	Specification of the simple and complex types that are used to represent the information of the products and auxiliary file types independently of the GOCE instrument to which is associated.

1.3. Acronyms and Abbreviations

DBL	Datablock
DFDL	Data Format Description Language
ENVISAT	ENVIronmental SATellite
GNU	GNU is Not Unix
HDR	Header
HK/TM	Housekeeping/Telemetry
MPH	Main Product Header
PDS	Payload Data Segment
SPH	Specific Product Header
W3C	World Wide Web Consortium
XML	eXtensible Mark-up Language

2. Target of preservation

GOCE L0 products in native format are available in tar/gzip format, with extension ".TGZ" (GNU-zipped tar file merging). However, the targets of preservation considered in this SAFE specialisation are the header (HDR) and datablock (DBL) parts of those products, i.e. the files which are stored within the tar/gzip files.

As a consequence, 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 following table summarises the list of product types in scope and points the reader to the sections in the document where the information about the main structure of the file can be found:

Product Types	Target of preservation	Structure specification
DFC_Anw_0_	Header file (.HDR) Datablock file (.DBL)	For HDR see section 3.1.1 For DBL see section 3.2.1.1
DFC_F01_0_	Header file (.HDR) Datablock file (.DBL)	For HDR see section 3.1.1 For DBL see section 3.2.1.1
DFC_F10_0_	Header file (.HDR) Datablock file (.DBL)	For HDR see section 3.1.1 For DBL see section 3.2.1.1
EGG_AUX_0_	Header file (.HDR) Datablock file (.DBL)	For HDR see section 3.1.1 For DBL see section 3.2.1.1
EGG_ICM_0_	Header file (.HDR) Datablock file (.DBL)	For HDR see section 3.1.1 For DBL see section 3.2.1.1
EGG_NOM_0_	Header file (.HDR) Datablock file (.DBL)	For HDR see section 3.1.1 For DBL see section 3.2.1.1
EGG_TOT_0_	Header file (.HDR) Datablock file (.DBL)	For HDR see section 3.1.1 For DBL see section 3.2.1.1
GRF_LOR_0_	Header file (.HDR) Datablock file (.DBL)	For HDR see section 3.1.1 For DBL see section 3.2.1.1
SST_ICB_0_	Header file (.HDR) Datablock file (.DBL)	For HDR see section 3.1.1 For DBL see section 3.2.1.1
SST_NOM_0_	Header file (.HDR) Datablock file (.DBL)	For HDR see section 3.1.1 For DBL see section 3.2.1.1
SST_TOT_0_	Header file (.HDR) Datablock file (.DBL)	For HDR see section 3.1.1 For DBL see section 3.2.1.1
STR_VC2_0_	Header file (.HDR) Datablock file (.DBL)	For HDR see section 3.1.1 For DBL see section 3.2.1.1
STR_VC3_01	Header file (.HDR) Datablock file (.DBL)	For HDR see section 3.1.1 For DBL see section 3.2.1.1
STR_VC3_02	Header file (.HDR) Datablock file (.DBL)	For HDR see section 3.1.1 For DBL see section 3.2.1.1
STR_VC3_03	Header file (.HDR) Datablock file (.DBL)	For HDR see section 3.1.1 For DBL see section 3.2.1.1
AUX_NOM_0_	Header file (.HDR) Datablock file (.DBL)	For HDR see section 3.1.1 For DBL see section 3.2.1.2

Table 1: Product Types Specification Index

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 L0 products and auxiliary files for both HDR and DBL files.

HDR files are simple/pure XML files (i.e. text files), without binary content that can be represented using standard XML Schemas. On the contrary, DBL files are not in XML files but rather binary files which, nevertheless, follow a structure which is represented by XML Schemas with DFDL annotations.

The representation information for both HDR and DBL files is described by mean 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 complex/simple types that are common to HDR and DBL files respectively.

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

3.1. Data Structures common to HDR files

The next figure provides a high level overview of the complex structures and basic types used to represent the information of L0 HDR products and auxiliary files in HDR format:

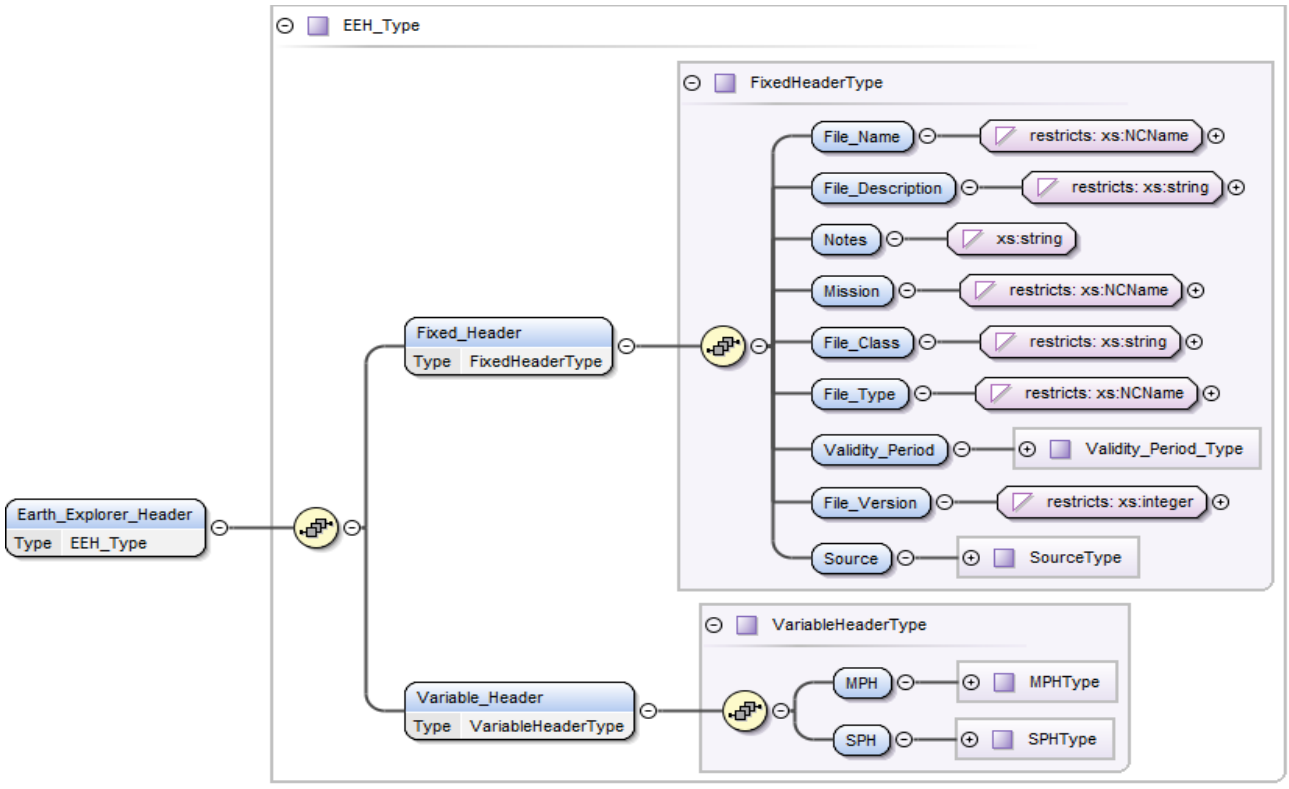

Figure 1: DFDL Schema organisation for HDR L0 products and auxiliary files

A detailed description of each complex type used for the representation information of these files is given below.

3.1.1. Root Element

This is the root XML element of the representation information for L0 HDR products and auxiliary files:

#	Name/Description	Format
1	**Earth_Explorer_Header** The XML Header file contains information identifying the product and easy to read as based on a standard syntax accessed by common tools available for visualising its content. The XML syntax has been chosen for the scope of the PDS. The XML Header file is composed by: * a Fixed Header * a Variable Header The Fixed Header is the common header for all files in the GOCE Ground Segment. That means it is applied to all files flowing amongst the sub-systems composing the PDS. The Variable Header is the header with format and content depending on the file type and kind of product.	EEH_Type

Table 2: Earth_Explorer_Header Specification

3.1.2. Simple Types

The DFDL schemas used to represent the information of the L0 products and auxiliary files make use of the standard W3C simple types (e.g. xs:string, xs:integer, xs:NCName, etc…).

Some of these types have been restricted for GOCE needs, resulting in new specific types detailed below.

3.1.2.1. ShortTimeType

Base Type	Format
xs:string	UTC=yyyy-mm-ddThh:mm:ss

Table 3: ShortTimeType Specification

3.1.2.2. LongTimeType

Base Type	Format
xs:string	UTC=yyyy-mm-ddThh:mm:ss.uuuuuu

Table 4: LongTimeType Specification

3.1.2.3. RestrictedSizeType

Base Type	Format
xs:integer	units: bytes Total Digits : "21"

Table 5: RestrictedSizeType Specification

3.1.2.4. real_s08_6f

Base Type	Format
xs:decimal	Total Digits : "8" Fraction Digits: "6"

Table 6: real_s08_6f Specification

3.1.2.5. PositionType

Base Type	Format
xs:decimal	Total Digits : "10" Fraction Digits: "3"

Table 7: PositionType Specification

3.1.2.6. VelocityType

Base Type	Format
xs:decimal	Total Digits : "10" Fraction Digits: "6"

Table 8: VelocityType Specification

3.1.2.7. Rel_Time_Asc_NodeType

Base Type	Format
xs:decimal	Total Digits : "10" Fraction Digits: "6"

Table 9: Rel_Time_Asc_NodeType Specification

3.1.2.8. LatLonType

Base Type	Format
xs:integer	Total Digits : "10"

Table 10: LatLonType Specification

3.1.3. Complex types

The following complex types are used by the DFDL schemas to represent the information of the L0 HDR products and auxiliary files:

3.1.3.1. EEH_Type

#	Name/Description	Format
1	**Fixed_Header**	FixedHeaderType
2	**Variable_Header**	VariableHeaderType

Table 11: EEH_Type Specification

3.1.3.2. FixedHeaderType

The Fixed Header is the common header for all files in the GOCE Ground Segment. That means it is applied to all files flowing amongst the sub-systems composing the PDS.

#	Name/Description	Format
1	**File_Name** Product File Name without the extension	xs:NCName Max Length : 55 bytes
2	**File_Description** This field shall contain a description of file product. Possible values: EGG Level-0 Product SST Level-0 Product EGG Level-0 TOT Product SST Level-0 TOT Product SST Calibration Product EGG Calibration Level-0 Product EGG Proof-mass calibration Product DFAC 1Hz Level 0 Product DFAC 10Hz Level 0 Product Star Tracker Level 0 Product CDMU Level 0 Product Auxiliary Level 0 Product EGG Auxiliary Product	xs:string
3	**Notes** This field shall be always empty	xs:string
4	**Mission** This field shall be always GOCE Possible values: GOCE	xs:NCName
5	**File_Class** This field is part of the File Name and indicates the type of processing. Possible values: OPER TEST	xs:string

#	Name/Description	Format
	OFFL RPRO	
6	**File_Type** This field is part of the File Name Possible values: EGG_TOT_0_ EGG_NOM_0_ EGG_ICM_0_ EGG_AUX_0_ STR_VC2_0_ STR_VC3_01 STR_VC3_02 STR_VC3_03 GRF_LOR_0_ SST_TOT_0_ SST_NOM_0_ SST_ICB_0_ DFC_F01_0_ DFC_F10_0_ DFC_A1X_0_ DFC_A1Y_0_ DFC_A1Z_0_ DFC_A2X_0_ DFC_A2Y_0_ DFC_A2Z_0_ DFC_A3X_0_ DFC_A3Y_0_ DFC_A3Z_0_ DFC_A4X_0_ DFC_A4Y_0_ DFC_A4Z_0_ DFC_A5X_0_ DFC_A5Y_0_ DFC_A5Z_0_ DFC_A6X_0_ DFC_A6Y_0_ DFC_A6Z_0_ AUX_NOM_0_	xs:NCName
7	**Validity_Period**	Validity_Period_Type
8	**File_Version** This field is version number of the generation of the product. It shall start from 0001 and increased by one anytime the same product shall be regenerated	xs:integer Total Digits : 4
9	**Source**	SourceType

Table 12: FixedHeaderType Specification

3.1.3.3. Validity_Period_Type

#	Name/Description	Format
1	**Validity_Start** For Product files, this field correspond to: * UTC Sensing Start Time of Level 0 product * UTC Start Time of Calibration activity For Auxiliary files, this field correspond to: * Minimum UTC Sensing Start Time of the Hk/TM data contained into the product	ShortTimeType
2	**Validity_Stop** For Product files, this field correspond to: * UTC Sensing Stop Time of Level 0 product * UTC Stop Time of Calibration activity For Auxiliary files, this field correspond to: * Maximum UTC Sensing Stop Time of Hk/TM data contained into the product	ShortTimeType

Table 13: Validity_Period_Type Specification

3.1.3.4. SourceType

#	Name/Description	Format
1	**System** Name of the Ground Segment component creating the product. Possible values: PDS	xs:NCName
2	**Creator** Name of the Ground Segment tool creating the product. Possible values: L0-Processor	xs:NCName
3	**Creator_Version** This field gives the version of the creator tool as VV.rr;	xs:decimal Total Digits : 4 Fraction Digits: 2
4	**Creation_Date** This field gives the UTC date of the generation of the file	ShortTimeType

Table 14: SourceType Specification

3.1.3.5. VariableHeaderType

#	Name/Description	Format
1	**MPH**	MPHType
2	**SPH**	SPHType

Table 15: VariableHeaderType Specification

3.1.3.6. MPHType

#	Name/Description	Format
1	**Product** Product File Name Note: the file name shall be without the extension.	xs:NCName Max Length : 55 bytes
2	**Proc_Stage_Code** Processing stage code Possible values: OPER TEST RPRO	xs:NCName
3	**Ref_Doc** Reference DFCB Document describing the product	xs:NCName Max Length : 23 bytes
4	**Proc_Time** Processing Time (Product Generation Time)	LongTimeType
5	**Software_Version** Processor Name and software version number. ProcessorName/VV.rr	xs:string Max Length : 14 bytes
6	**Phase** Phase Code: If not used set to X	xs:string Max Length : 1 bytes
7	**Cycle** Cycle number. If not used set to 000	xs:integer Total Digits : 3
8	**Rel_Orbit_Start** Relative Orbit Number at sensing start time. If not used set to 00000	xs:integer Total Digits : 5
9	**Abs_Orbit_Start** Absolute Orbit Number at sensing start time. If not used set to 000000	xs:integer Total Digits : 6
10	**Rel_Orbit_Stop** Relative Orbit Number at sensing stop time. If not used set to 00000	xs:integer Total Digits : 5
11	**Abs_Orbit_Stop** Absolute Orbit Number at sensing stop time. If not used set to 00000	xs:integer Total Digits : 6
12	**State_Vector_Time** UTC state vector time	LongTimeType
13	**Delta_UT1** Universal Time Correction: DUT1 = UT1 – UTC If not used set to +.000000	Delta_UT1_Type
14	**X_Position** X position in Earth Fixed Reference If not used set to +0000000.000	RestrictedPositionType
15	**Y_Position** Y position in Earth Fixed Reference	RestrictedPositionType

#	Name/Description	Format
	If not used set to +0000000.000	
16	**Z_Position** Z position in Earth Fixed Reference If not used set to +0000000.000	RestrictedPositionType
17	**X_Velocity** X velocity in Earth Fixed Reference If not used set to +0000.000000	RestrictedVelocityType
18	**Y_Velocity** Y velocity in Earth Fixed Reference If not used set to +0000.000000	RestrictedVelocityType
19	**Z_Velocity** Z velocity in Earth Fixed Reference If not used set to +0000.000000	RestrictedVelocityType
20	**State_Vector_Source** Source of Orbit State Vector Record FP = FOS predicted Possible values: FP	xs:NCName
21	**Product_Err** 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 For the Auxiliary L0 product set always to 0	xs:integer
22	**Tot_Size** Total size of the product	SizeType

Table 16: MPHType Specification

3.1.3.7. SizeType

#	Name/Description	Format
1	Size Type	RestrictedSizeType **Attribute:** Name: "unit" Type: "xs:NCName" Use : "required"

Table 17: SizeType Specification

3.1.3.8. Delta_UT1_Type

#	Name/Description	Format
1	Delta UT1 Type	real_s08_6f **Attribute:**

#	Name/Description	Format
		Name: "unit" Type: "xs:string" Use : "required"

Table 18: Delta_UT1_Type Specification

3.1.3.9. RestrictedPositionType

#	Name/Description	Format
1	Restricted Position Type	PositionType **Attribute:** Name: "unit" Type: "xs:NCName" Use : "required"

Table 19: RestrictedPositionType Specification

3.1.3.10. RestrictedVelocityType

#	Name/Description	Format
1	Restricted Velocity Type	VelocityType **Attribute:** Name: "unit" Type: "xs:string" Use : "required"

Table 20: RestrictedVelocityType Specification

3.1.3.11. SPHType

#	Name/Description	Format
1	**SPH_Descriptor** Name describing the Specific Product Header Possible values: EGG_TOT_0_ SPECIFIC HEADER EGG_NOM_0_ SPECIFIC HEADER DFC_F01_0_ SPECIFIC HEADER DFC_F10_0_ SPECIFIC HEADER STR_VC2_0_ SPECIFIC HEADER STR_VC3_01 SPECIFIC HEADER STR_VC3_02 SPECIFIC HEADER STR_VC3_03 SPECIFIC HEADER GRF_LOR_0_ SPECIFIC HEADER EGG_ICM_0_ SPECIFIC HEADER SST_TOT_0_ SPECIFIC HEADER SST_NOM_0_ SPECIFIC HEADER SST_ICB_0_ SPECIFIC HEADER DFC_A1X_0_ SPECIFIC HEADER DFC_A1Y_0_ SPECIFIC HEADER DFC_A1Z_0_ SPECIFIC HEADER	xs:string

#	Name/Description	Format
	DFC_A2X_0_ SPECIFIC HEADER DFC_A2Y_0_ SPECIFIC HEADER DFC_A2Z_0_ SPECIFIC HEADER DFC_A3X_0_ SPECIFIC HEADER DFC_A3Y_0_ SPECIFIC HEADER DFC_A3Z_0_ SPECIFIC HEADER DFC_A4X_0_ SPECIFIC HEADER DFC_A4Y_0_ SPECIFIC HEADER DFC_A4Z_0_ SPECIFIC HEADER DFC_A5X_0_ SPECIFIC HEADER DFC_A5Y_0_ SPECIFIC HEADER DFC_A5Z_0_ SPECIFIC HEADER DFC_A6X_0_ SPECIFIC HEADER DFC_A6Y_0_ SPECIFIC HEADER DFC_A6Z_0_ SPECIFIC HEADER AUX_NOM_0_ SPECIFIC HEADER EGG_AUX_0_ SPECIFIC HEADER	
2	**Sensing_Start** UTC start time of data sensing.	LongTimeType Min Occurs : 0
3	**Sensing_Stop** UTC stop time of data sensing.	LongTimeType Min Occurs : 0
4	**Rel_Time_ASC_Node_Start** Relative time since crossing ascending node time relative to start time of data sensing.	RestrictedRel_Time_Asc_NodeType Min Occurs : 0
5	**Rel_Time_ASC_Node_Stop** Time of the ascending node relative to stop time of data sensing. Relative time since crossing ascending node time relative to stop time of data sensing.	RestrictedRel_Time_Asc_NodeType Min Occurs : 0
6	**Equator_Cross_Time** Time of equator crossing at the ascending node relative to the sensing start time.	LongTimeType Min Occurs : 0
7	**Equator_Cross_Long** Longitude of equator crossing at the ascending node relative to the sensing start time (positive East, 0 = Greenwich) referred to WGS84.	RestrictedLatLonType Min Occurs : 0
8	**Ascending_Flag** Orbit orientation at the sensing start time. Ascending (A) or Descending (D) Possible values: A D	xs:string Min Occurs : 0
9	**Product_Location**	Product_Location_Type Min Occurs : 0
10	**Product_Conf_Data**	Product_Conf_Data_Type

#	Name/Description	Format
		Min Occurs : 0
11	**DSDs**	DSDs_Type

Table 21: SPHType Specification

3.1.3.12. Product_Location_Type

#	Name/Description	Format
1	**Start_Lat** Latitude of first satellite nadir point at the Sensing Start time (positive North)	RestrictedLatLonType
2	**Start_Long** Longitude of first satellite nadir point at the Sensing Start time (positive East, 0 = Greenwich)	RestrictedLatLonType
3	**Stop_Lat** Latitude of first satellite nadir point at the Sensing Stop time (positive North)	RestrictedLatLonType
4	**Stop_Long** Longitude of first satellite nadir point at the Sensing Stop time (positive East, 0 = Greenwich)	RestrictedLatLonType

Table 22: Product_Location_Type Specification

3.1.3.13. Product_Conf_Data_Type

#	Name/Description	Format
1	**Num_ISPs** Number of ISPs in the Level 0	xs:integer Total Digits : 7
2	**Num_Missing_ISPs** Number of missing ISPs	xs:integer Total Digits : 7
3	**Num_Error_ISPs** Number of ISPs containing CRC errors	xs:integer Total Digits : 7
4	**Num_Discarded_ISPs** Number of ISPs discarded	xs:integer Total Digits : 7
5	**Num_RS_ISPs** Number of ISPs with Reed-Solomon correction in the Level 0.	xs:integer Total Digits : 7
6	**Num_RS_Corrections** Number of symbols corrected with Reed-Solomon in the product.	xs:integer Total Digits : 7

Table 23: Product_Conf_Data_Type Specification

3.1.3.14. DSDs_Type

#	Name/Description	Format
1	**List_of_DSDs** Number of Data Sets	List_of_DSDs_Type

Table 24: DSDs_Type Specification

3.1.3.15. List_of_DSDs_Type

Attribute:

Name	Use	Type
count	required	xs:integer

#	Name/Description	Format
1	**Data_Set_Descriptor**	Data_Set_DescriptorType Max Occurs : unbounded

Table 25: List_of_DSDs_Type Specification

3.1.3.16. RestrictedRel_Time_Asc_NodeType

#	Name/Description	Format
1	Restricted Rel Time Ascending Node Type	Rel_Time_Asc_NodeType **Attribute:** Name: "unit" Type: "xs:NCName" Use : "required"

Table 26: RestrictedRel_Time_Asc_NodeType Specification

3.1.3.17. RestrictedLatLonType

#	Name/Description	Format
1	Restricted Latitude/Longitude Type	LatLonType **Attribute:** Name: "unit" Type: "xs:string" Use : "required"

Table 27: RestrictedLatLonType Specification

3.1.3.18. Data_Set_DescriptorType

#	Name/Description	Format
1	**Data_Set_Name** Name describing the Data Set	xs:string Min Occurs : 0 Max Length : 28 bytes
2	**Data_Set_Type** Type of Data Set. Measurement (M) or Reference (R) Possible values: M	xs:NCName Min Occurs : 0

#	Name/Description	Format
	R	
3	**File_Name** Name of Reference File	xs:string Min Occurs : 0 Max Length : 62 bytes
4	**Data_Set_Offset** Offset in bytes from the beginning of the file (MPH+SPH including DSD) This field will be filled only for measurement Data Set.	SizeType Min Occurs : 0
5	**Data_Set_Size** Size of the Data Set This field will be filled only for measurement Data Set	SizeType Min Occurs : 0
6	**Num_of_Records** Number of records in the Data Set (filled only for measurements Data Set)	xs:integer Min Occurs : 0 Total Digits : 11
7	**Record_Size** Size in bytes of a record	SizeType Min Occurs : 0
8	**Byte_Order** Byte ordering information. It describes the endianess of the data set. 3210 (Big-endian) 0123 (Little-endian) For the Reference DSD this field is empty	xs:unsignedShort Min Occurs : 0 Total Digits : 4

Table 28: Data_Set_DescriptorType Specification

3.2. Data Structures common to DBL files

The data structure of DBL products and auxiliary files is different in this case given that the SPH complex type varies from one file type to another.

The next figure provides a high level overview of the complex structures and basic types used to represent the information of L0 products in DBL format:

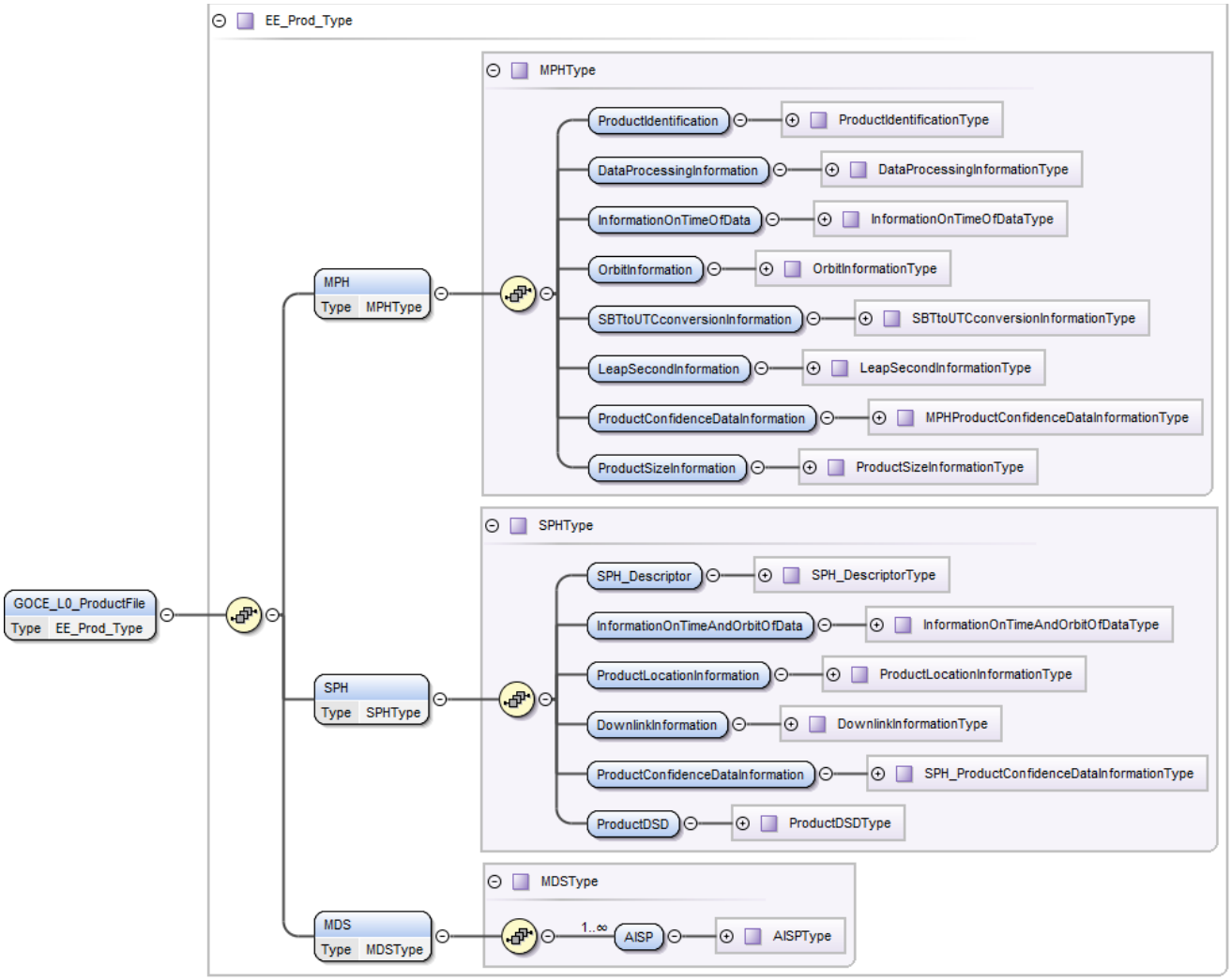

Figure 2: DFDL schema organisation for DBL L0 product files

The next figure provides a high level overview of the complex structures and basic types used to represent the information of L0 auxiliary files in DBL format:

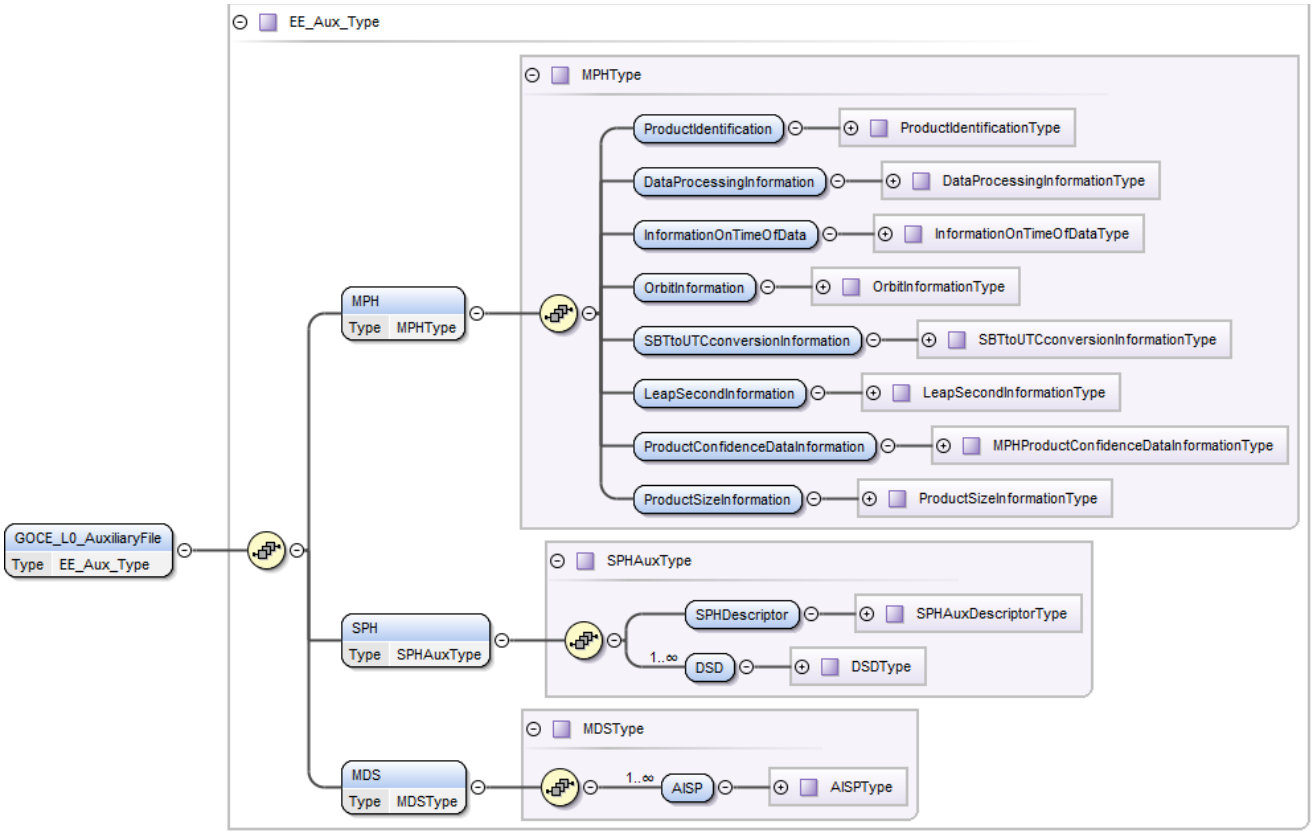

Figure 3: DFDL schema organisation for DBL L0 Auxiliary files

A detailed description of each complex type used for the representation information of these files is given in the next sub-sections.

3.2.1. Root Elements

Considering the different data structures between products and auxiliary files, two different root nodes have been specified for L0 DBL files:

3.2.1.1. Product files

This is the root element of the representation information for DBL L0 product files.

#	Name/Description	Format
1	**GOCE_L0_ProductFile** GOCE Level-0 Product File 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 is completely binary and contains one GOCE Measurement Data Set (MDS) for each file.	EE_Prod_Type

#	Name/Description	Format
	It has to be remarked that binary data written in the Level-0 must follow the big-endian representation.	

Table 29: GOCE_L0_ProductFile Specification

3.2.1.2. Auxiliary files

This is the root element of the representation information for DBL L0 auxiliary files:

#	Name/Description	Format
1	**GOCE_L0_AuxiliaryFile** GOCE Level-0 Auxiliary file The file structure will follow the one defined for the GOCE level-0 products. Each auxiliary level-0 file is composed by: * Main Product Header (MPH) * Specific Product Header (SPH) * One or more Measurement Data Blocks The MPH and SPH blocks are ASCII, whereas the MDSs are completely binary. Each MDS contains one HK/TM telemetry type. The number of MDS is variable and depends on the number of telemetries extracted from the VC3 telemetry file. The list of telemetry contained into the auxiliary level 0 product is specified into the AUX_VC3_TM_ file. It has to be remarked that binary data written in the Level-0 must follow the big-endian representation.	EE_Aux_Type

Table 30: GOCE_L0_AuxiliaryFile Specification

3.2.2. Simple Types

The DFDL schemas used to represent the information of the L0 products and auxiliary files make use of the standard W3C simple types (e.g. xs:string, xs:integer, xs:NCName, etc…).

Some of these types have been restricted for GOCE needs, resulting in new specific types detailed below.

3.2.2.1. uc

Base Type	Length (bytes)	Comments
xs:string	1	

Table 31: uc Specification

3.2.2.2. 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 32: utc Specification

3.2.2.3. int_s04d

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

Table 33: int_s04d Specification

3.2.2.4. int_s06d

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

Table 34: int_s06d Specification

3.2.2.5. int_06d

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

Table 35: int_06d Specification

3.2.2.6. int_s07d

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

Table 36: int_s07d Specification

3.2.2.7. int_07d

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

Table 37: int_07d Specification

3.2.2.8. int_s11d

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

Table 38: int_s11d Specification

3.2.2.9. int_s21d

Base Type	Length (bytes)	Comments
xs:unsignedLong	21	Pattern : '+'000000000000000000;-000000000000000000

Table 39: int_s21d Specification

3.2.2.10. real_s08_6f

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

Table 40: real_s08_6f Specification

3.2.2.11. real_s11_6f

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

Table 41: real_s11_6f Specification

3.2.2.12. real_11_6f

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

Table 42: real_11_6f Specification

3.2.2.13. real_s12_3f

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

Table 43: real_s12_3f Specification

3.2.2.14. real_s12_6f

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

Table 44: real_s12_6f Specification

3.2.3. Complex Types

The following complex types are used by the DFDL schemas used to represent the information of the L0 DBL products and auxiliary files:

3.2.3.1. Product files

These complex types have been defined specifically for product files and are not used to represent auxiliary files.

3.2.3.1.1.EE_Prod_Type

#	Name/Description	Format
1	**MPH**	MPHType
2	**SPH**	SPHType
3	**MDS**	MDSType

Table 45: EE_Prod_Type Specification

3.2.3.1.2.SPHType

#	Name/Description	Format
1	**SPH_Descriptor**	SPH_DescriptorType
2	**InformationOnTimeAndOrbitOfData**	InformationOnTimeAndOrbitOfDataType
3	**ProductLocationInformation**	ProductLocationInformationType
4	**DownlinkInformation**	DownlinkInformationType
5	**ProductConfidenceDataInformation**	SPH_ProductConfidenceDataInformationType
6	**ProductDSD**	ProductDSDType

Table 46: SPHType Specification

3.2.3.1.3.SPH_DescriptorType

#	Name/Description	Format
1	**SPH_DESCRIPTOR="**	xs:string 16 bytes
	ASCII string describing the name of the Specific Product Header.	xs:string 28 bytes
	"	uc 1 bytes

Table 47: SPH_DescriptorType Specification

3.2.3.2. Auxiliary files

These complex types have been defined specifically for auxiliary files and are not used to represent product files.

3.2.3.2.1.EE_Aux_Type

#	Name/Description	Format
1	**MPH**	MPHType
2	**SPH**	SPHAuxType
3	**MDS**	MDSType

Table 48: EE_Aux_Type Specification

3.2.3.2.2.SPHAuxType

#	Name/Description	Format
1	**SPHDescriptor**	SPHAuxDescriptorType
2	**DSD**	DSDType

#	Name/Description	Format
	* Data Set Descriptor for the Measurement Data Set #[1..n) * Data Set Descriptor of the VC3 Telemetry file used for generating the product (Reference DSD) * Data Set Descriptor for the Orbit State Vector File used (Reference DSD) * Data Set Descriptor for the OBT-UTC correlation file if used (Reference DSD) * Data Set Descriptor for List of Telemetry Types used (Reference DSD) * Data Set Descriptor for the Auxiliary Processor Configuration File used (Reference DSD)	Min Occurs : 1 Max Occurs : unbounded

Table 49: SPHAuxType Specification

3.2.3.2.3.SPHAuxDescriptorType

#	Name/Description	Format
1	**SPH_DESCRIPTOR="**	xs:string 16 bytes
	ASCII string describing the name of the Specific Product Header.	xs:string 28 bytes
	"	uc 1 bytes
2	**Spare**	xs:string 51 bytes

Table 50: SPHAuxDescriptorType Specification

3.2.3.3. Product & Auxiliary files

These complex types have been defined to represent both products and auxiliary files.

3.2.3.3.1.MPHType

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

Table 51: MPHType Specification

3.2.3.3.2.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 O = Off-Line S = Special product T = Test Product V = fully validated (fully consolidated) 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	xs:string 23 bytes
	"	uc 1 bytes
4	**Spare**	xs:string 40 bytes

Table 52: ProductIdentificationType Specification

3.2.3.3.3.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 The remaining characters are filled by blanks Possible values: Kiruna Svalbard	xs:string 20 bytes
	"	uc 1 bytes
2	**PROC_CENTER="**	xs:string

#	Name/Description	Format
		13 bytes
	Processing Center ID code Possible values: PDS	xs:string 6 bytes
	"	uc 1 bytes
3	**PROC_TIME="**	xs:string 11 bytes
	Processing Time (Product Generation Time)	utc
	"	uc 1 bytes
4	**SOFTWARE_VER="**	xs:string 14 bytes
	Processor name, up to 8 characters, and software version number followed by trailer blanks if any.	xs:string 14 bytes
	"	uc 1 bytes
5	**Spare**	xs:string 40 bytes

Table 53: DataProcessingInformationType Specification

3.2.3.3.4.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. If not used set to 27 blanks	utc
	"	uc 1 bytes
2	**SENSING_STOP="**	xs:string 14 bytes
	UTC stop time of data sensing If not used set to 27 blanks	utc
	"	uc 1 bytes
3	**Spare**	xs:string 40 bytes

Table 54: InformationOnTimeOfDataType Specification

3.2.3.3.5.OrbitInformationType

Information on GOCE 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	int_s04d
3	**REL_ORBIT_START=**	xs:string 16 bytes
	Relative Orbit Number at sensing start time. If not used set to +00000	int_s06d
4	**ABS_ORBIT_START=**	xs:string 16 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 to 27 blanks.	utc
	"	uc 1 bytes
6	**DELTA_UT1=**	xs:string 10 bytes
	Universal Time Correction: value<s> DUT1 = UT1 – UTC Not used for GOCE. It shall be 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

#	Name/Description	Format
	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	**STATE_VECTOR_SOURCE="**	xs:string 21 bytes
	Source Record of Orbit State Vector	xs:string 2 bytes
	"	uc 1 bytes
14	**Spare**	xs:string 40 bytes

Table 55: OrbitInformationType Specification

3.2.3.3.6.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
	Not used for GOCE and it shall be set to 27 blanks	xs:string 27 bytes
	"	uc 1 bytes
2	**SAT_BINARY_TIME=**	xs:string 16 bytes
	Satellite Binary Time Not used for GOCE and it shall be set to zeros	xs:string 11 bytes
3	**CLOCK_STEP=**	xs:string 11 bytes
	Clock Step: value<ps> Not used for GOCE and it shall be set to 00000000000<ps>	xs:string 15 bytes
4	**Spare**	xs:string 32 bytes

Table 56: SBTtoUTCconversionInformationType Specification

3.2.3.3.7.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 occurrence (i.e.: midnight of the day after the leap second).	xs:string 27 bytes
	"	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. This field is always set to 0 considering that GOCE products have true UTC times	uc
4	**Spare**	xs:string 40 bytes

Table 57: LeapSecondInformationType Specification

3.2.3.3.8.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 58: MPHProductConfidenceDataInformationType Specification

3.2.3.3.9.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	**CRC=**	xs:string 4 bytes
	Cyclic Redundancy Code computed as overall value of all records of the Measurement Data Set.	int_s06d
7	**Spare**	xs:string 29 bytes

Table 59: ProductSizeInformationType Specification

3.2.3.3.10.InformationOnTimeAndOrbitOfDataType

#	Name/Description	Format
1	**SENSING_START_UTC="**	xs:string 19 bytes
	UTC start time of data sensing.	utc
	"	uc 1 bytes
2	**SENSING_STOP_UTC="**	xs:string 18 bytes
	UTC stop time of data sensing.	utc
	"	uc 1 bytes
3	**ABS_ORBIT_START=**	xs:string 16 bytes

#	Name/Description	Format
	Absolute Orbit Number at sensing start time.	int_06d
4	**ABS_ORBIT_STOP=**	xs:string 15 bytes
	Absolute Orbit Number at sensing stop time.	int_06d
5	**REL_ORBIT_START=**	xs:string 16 bytes
	Relative Orbit Number at sensing start time.	int_06d
6	**REL_ORBIT_STOP=**	xs:string 15 bytes
	Relative Orbit Number at sensing stop time.	int_06d
7	**REL_TIME_ASC_NODE_START=**	xs:string 24 bytes
	Relative time since crossing ascending node time relative to start time of data sensing: value<s>	xs:string 14 bytes
8	**REL_TIME_ASC_NODE_STOP=**	xs:string 23 bytes
	Relative time since crossing ascending node time relative to stop time of data sensing: value<s>	xs:string 14 bytes
9	**EQUATOR_CROSS_TIME_UTC="**	xs:string 24 bytes
	Time of equator crossing at the ascending node of the sensing start time.	utc
	"	uc 1 bytes
10	**EQUATOR_CROSS_LONG=**	xs:string 19 bytes
	Longitude of equator crossing at the ascending node of the sensing start time (positive East, 0 = Greenwich) referred to WGS84: value<10-6degN>	xs:string 21 bytes
11	**ASCENDING_FLAG=**	xs:string 15 bytes
	Orbit orientation at the sensing start time. Ascending (A) or Descending (D) Possible values: A D	uc

Table 60: InformationOnTimeAndOrbitOfDataType Specification

3.2.3.3.11.ProductLocationInformationType

#	Name/Description	Format
1	**START_LAT=**	xs:string 10 bytes
	Latitude of first satellite nadir point at the Sensing Start time (positive North) referred to WGS84: value<10-6degN>	xs:string 21 bytes
2	**START_LONG=**	xs:string 11 bytes

#	Name/Description	Format
	Longitude of first satellite nadir point at the Sensing Start time (positive East, 0 = Greenwich) referred to WGS84: value<10-6degN>	xs:string 21 bytes
3	**STOP_LAT=**	xs:string 9 bytes
	Latitude of first satellite nadir point at the Sensing Stop time (positive North) referred to WGS84: value<10-6degN>	xs:string 21 bytes
4	**STOP_LONG=**	xs:string 10 bytes
	Longitude of first satellite nadir point at the Sensing Stop time (positive East, 0 = Greenwich) referred to WGS84: value<10-6degN>	xs:string 21 bytes

Table 61: ProductLocationInformationType Specification

3.2.3.3.12.DownlinkInformationType

#	Name/Description	Format
1	**DOWNLINK_START_UTC="**	xs:string 20 bytes
	UTC start time of data downlink. For Goce the downlink time is referred to the start processing time of the Level0 product.	utc
	"	uc 1 bytes
2	**DOWNLINK_STOP_UTC="**	xs:string 19 bytes
	UTC stop time of data downlink. For Goce the downlink time is referred to the stop processing time of the Level0 product.	utc
	"	uc 1 bytes
3	**DOWNLINK_ORBIT=**	xs:string 15 bytes
	Absolute Orbit Number at downlink start time.	int_06d

Table 62: DownlinkInformationType Specification

3.2.3.3.13.SPH_ProductConfidenceDataInformationType

#	Name/Description	Format
1	**NUM_ISPS=**	xs:string 9 bytes
	Number of ISPs in the Level 0	int_07d
2	**NUM_MISSING_ISPS=**	xs:string 17 bytes
	Number of missing ISPs	int_07d
3	**NUM_ERROR_ISPS=**	xs:string 15 bytes

#	Name/Description	Format
	Number of ISPs containing CRC errors.	int_07d
4	**NUM_DISCARDED_ISPS=**	xs:string 19 bytes
	Number of ISPs discarded	int_07d
5	**NUM_RS_ISPS=**	xs:string 12 bytes
	Number of ISPs with Reed-Solomon correction in the Level 0.	int_07d
6	**NUM_RS_CORRECTIONS=**	xs:string 19 bytes
	Number of symbols corrected with Reed-Solomon.	int_07d
7	**Spare**	xs:string 72 bytes

Table 63: SPH_ProductConfidenceDataInformationType Specification

3.2.3.3.14.ProductDSDType

#	Name/Description	Format
1	**DSD_MDS** Data Set Descriptor for the Measurement Data Set (MDS)	DSDType Min Occurs : 1 Max Occurs : 1
2	**DSD_PCF** Data Set Descriptor for the Processor Configuration File used (Reference DSD)	DSDType Min Occurs : 1 Max Occurs : 1
3	**DSD_TEL** Data Set Descriptor of the Telemetry file used or file used for generating the product (Reference DSD)	DSDType Min Occurs : 1 Max Occurs : 1
4	**DSD_OSV** Data Set Descriptor for the Orbit State Vector File used (Reference DSD)	DSDType Min Occurs : 1 Max Occurs : 1
5	**DSD_OBT** Data Set Descriptor for the OBT-UTC correlation file if used (Reference DSD)	DSDType Min Occurs : 0 Max Occurs : 1
6	**DSD_VC3** Data Set Descriptor for VC3 Telemetry file if used (Reference DSD) otherwise Data Set Descriptor spare	DSDType Min Occurs : 0 Max Occurs : 1
7	**Spare**	xs:string

Table 64: ProductDSDType Specification

3.2.3.3.15.DSDType

#	Name/Description	Format
1	**DS_NAME="**	xs:string 9 bytes
	Name describing the Data Set Left justified and filled by trailer blanks up to the length.	xs:string 28 bytes

#	Name/Description	Format
	"	uc 1 bytes
2	**DS_TYPE=**	xs:string 8 bytes
	Type of the Data Set. Measurement (M) or Reference (R) Possible values: M R	uc
3	**FILENAME="**	xs:string 10 bytes
	Name of the Reference File. Used if DS_TYPE is set to R. It is left justified with trailer blanks Note: the file name shall be without the extension. If 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 00000000000000000000<bytes>	xs:string 28 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 65: DSDType Specification

3.2.3.3.16.MDSType

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

#	Name/Description	Format
		Max Occurs : unbounded

Table 66: MDSType Specification

3.2.3.3.17.AISPType

#	Name/Description	Format
1	**Annotation**	AnnotationType
2	**ISP**	InstrumentSourcePacketType

Table 67: AISPType Specification

3.2.3.3.18.AnnotationType

#	Name/Description	Format
1	**ISP_SENSING**	ISPSensingTimeType
2	**ISP_DOWNLINK**	ISPDownlinkTimeType
3	**ISP_PRODUCT_CONFIDENCE**	ISPProductConfidenceDataType

Table 68: AnnotationType Specification

3.2.3.3.19.ISPSensingTimeType

#	Name/Description	Format
1	**elapsedDays** Number of days elapsed since the 1st of January 2000 at 00:00:00 hour (units: days)	xs:int 4 bytes
2	**elapsedSeconds** Number of seconds elapsed since the beginning of that day (units: s)	xs:unsignedInt 4 bytes
3	**elapsedMicroSeconds** Number of microseconds elapsed since the last second (units: us)	xs:unsignedInt 4 bytes

Table 69: ISPSensingTimeType Specification

3.2.3.3.20.ISPDownlinkTimeType

#	Name/Description	Format
1	**elapsedDays** Number of days elapsed since the 1st of January 2000 at 00:00:00 hour (units: days)	xs:int 4 bytes
2	**elapsedSeconds** Number of seconds elapsed since the beginning of that day (units: s)	xs:unsignedInt 4 bytes
3	**elapsedMicroSeconds** Number of microseconds elapsed since the last second (units: us)	xs:unsignedInt 4 bytes

Table 70: ISPDownlinkTimeType Specification

3.2.3.3.21.ISPProductConfidenceDataType

#	Name/Description	Format
1	**packetLength** ISP Length = (length of source packet excluding 6	xs:unsignedShort 2 bytes

#	Name/Description	Format
	bytes) – 1. This field uses the same definition as the Packet Length field of the Packet Header as specified in CCSDS definitions.(units: bytes)	
2	**nVCDUs_ISP** Number of current ISP VCDUs containing current ISP	xs:unsignedShort 2 bytes
3	**nVCDUs_ISP_Reed_Solomon** Number of VCDUs containing current ISP that are corrected with Reed-Solomon	xs:unsignedShort 2 bytes
4	**nVCDUs_ISP_RS** Number of VCDUs containing current ISP that were incorrigible with RS	xs:unsignedShort 2 bytes
5	**nVCDUs_missing** Number VCDUs of missing containing current ISP	xs:unsignedShort 2 bytes
6	**nSymbols** Number of symbols corrected by RS in CADUs containing current ISP	xs:unsignedShort 2 bytes
7	**crc** CRC Error Flag, identifying the detection of CRC error in the packet (set to FF in case of error otherwise set to 00).	xs:unsignedByte 1 bytes
8	**spares1**	xs:unsignedByte 1 bytes
9	**spares2**	xs:unsignedByte 1 bytes
10	**spares3**	xs:unsignedByte 1 bytes

Table 71: ISPProductConfidenceDataType Specification

3.2.3.3.22.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 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	**packetHeader**	packetHeader_Type
2	**dataFieldHeader** 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.	xs:hexBinary 10 bytes
3	**applicationData** 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. Length = packetHeader/packetLength + 1 - dataFieldHeaderLength - packetErrorControlLength	xs:hexBinary {../packetHeader/packetLength + 1 - 10 - 2}
4	**packetErrorControl** The Packet Error Control field is an optional subdivision of the Packet Data Field. The standard mechanisms that are used by the ground telemetry acquisition systems to extract Telemetry Packets from the Transfer Frame are such that no use is made of a Packet Error Control field (Telemetry Packets do not have such a field in any case). However, at the discretion of the user, an optional error detection code may be appended to the Source Data Field so that the ultimate recipient of the data is able to verify that the integrity of the complete Source Packet structure has been preserved during the entire transport process. The length of the Packet Error Control field shall be a multiple (integer) of octets.	xs:hexBinary 2 bytes

Table 72: InstrumentSourcePacketType Specification

3.2.3.3.23.packetHeader_Type

#	Name/Description	Format
1	**packetID**	packetIDType
2	**packetSeqControl**	packetSeqControlType
3	**packetLength** The Packet Length is a 16-bit field which specifies the number of octets contained within the Packet Data Field. The number is a binary value 'C' expressed as follows: C = [(Number of octets in Data Field) - I] Therefore, it should be noted that the actual length of the entire Source Packet will implicitly be 6 octets longer, since the standard 48-bit Packet Header always precedes the Packet Data Field. Also, the smallest possible Source Packet length is 7 octets and the largest possible is 65 542 octets.	xs:unsignedShort 2 bytes

Table 73: packetHeader_Type Specification

3.2.3.3.24.packetIDType

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 3 bytes
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 1 bytes
3	**dataFieldHeaderFlag** 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 1 bytes
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.	xs:unsignedShort 11 bytes

#	Name/Description	Format
	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 associated 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 74: packetIDType Specification

3.2.3.3.25.packetSeqControlType

#	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 2 bytes
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 14 bytes

Table 75: packetSeqControlType Specification