EUMETSAT

Generic Product Format Specification

Doc.No.	:	EPS.GGS.SPE.96167
Issue	:	v7E
Date	:	30 November 2010

EUMETSAT
Am Kavalleriesand 31, D-64295 Darmstadt, Germany
Tel: +49 6151 807-7
Fax: +49 6151 807 555
http://www.eumetsat.int

Document Change Record

Issue / Revision	Date	DCN. No	Changed Pages / Paragraphs
2 Draft 2	23/07/99		First issue
3 Draft B	23/07/99		First Issue for EPS Ground Segment ITT
4 Draft A	28/06/00	EUM.EPS.SYS.DCN.036	LEO/C/TP • Completely revised in line with NRT record structure • Editorial changes to clarify text. • Addition of GTS product section. • Update of generic Level 0 product format
4 Draft B	15/01/00		LEO/C/TP • Moved all generic SPHR fields to the MPHR leaving SPHR solely for level/instrument specific issues • Moved record descriptions to Annex in Excel format • Added Section 2 – “Terminology” (section moved from PCD) • Removed empty section for Generic EPS Browse products. • Removed “validation” and “commissioning” as processing modes • Made INST_MODEL an enumerated field
4 Draft C	12/04/01		LEO/C/TP • Updated in accord with PGS/PFS Algorithm Panel Meeting with Alcatel (11/04/01) • Removed footer concept in product , including MPFR and SPFR records. Section 5 “Footer Section” deleted. • Amalgamated NRT and Full product formats into one single EPS product format • Inserted Sections 4 and 5 – “Global Auxiliary Data Section” and “Variable Auxiliary Data Section”. GAD records move into the Global Aux. Data Section from Header Section. VAD records moved into Variable Aux. Data Section from body section and grouped by subclass and time-ordered. • Generic Record Header: Replaced “Record Instance” with “Record Subclass” for better clarity of concept. Added Record_Time field. • Generic Record Header: Replaced “Defining Group” with “Instrument Group” for better clarity of concept • Introduced PDU definition which supersedes restrictions on products arriving at NRT terminals with complete records. This was therefore removed. • Updated SRC_DATA_TYPE definition to include “Not Applicable” • Added DISP_MODE to MPHR

			• MPHR and SPHR redefined to be in ASCII format. Relevant ASCII record concepts added to document. • Introduced SUBSETTED_PRODUCT and SUBSET_TIME to allow for UMARF generated subsetted products with unique names • Allow binary record field names to include lower case letters (to keep compatibility with IASI Level 1 naming conventions from CNES). • Removed MPHR fields: PRODUCT_CONF and PROC_QUAL • Removed TIME_SEQUENCE_CODE, EARTH_LOCATION_ERROR and EARTH_LOCATION_ERROR from MPHR • Added RECORD_START_TIME and RECORD_STOP_TIME to generic record headers • Satellite Data (X-band data) removed from Level 0 product (VIADRs) • Updated Level 0 MDR definition –combined MetOp and NOAA into one generic record • Defined new INST type of NOAA -used for NOAA level 0 products that contain all NOAA instruments in each product • Updated [RD-107] - removed [TBC] • Added additional PROD_TYPE for ASCAT of FUL. Updated SZO and SZR PROD_TYPE descriptions.
4 Draft D	24/06/01		LEO/C/TP • Added GRH details for records • Corrected Section numbering • Removed SRC_DATA_QUAL from MPHR. Non-generic field. If required, should be added to individual SPHRs • Removed GTS Section (GTS formatting covered by PGS documents) • Updated SOURCE_GR_STN enumerated values • Added captions to all tables • Removed BRx “Browse product” product type • Updated GEADR/VEADR subclass methodology • Updated Document Signature table
5.0	01/06/01		Initial version for CGS PDR release
5.1	01/06/01		LEO/C/TP • Updates for CGS PDR release • Various editorial corrections • Increase MPHR.PROC_CENTRE field to 4 characters. Amend values accordingly and replaced CGSx with CGS1 and CGS2 to identify different ground segments • Amended issues to do with ASCII-based records • Added new data type x-char in order to allow ASCII records to have additional characters not allowed by the char data type. • Rearranged order of sections to clarify the

			construction of products and to avoid
5.2	10/07/01	EUM.EPS.SYS.DCN.01.080	Includes the Annex which was missing from the hardcopy distribution of Issue 5 Rev 1.
6.0	18/02/02		LEO/C/TP • Added Section 7 Product Naming Convention (taken from Product Convention Document (PCD) EPS/SYS/TEN/990007 Issue 3 Rev 0) • Added Section 8 Product Format Version Control (taken from PCD Issue 3 Rev 0) • Removed PCD as an applicable document (PCD now has nothing to do with products). • Minor updates to text to account for above changes (e.g. references to PCD changed to internal references to Section 7 and Section 8) • Reworded Section 8 to clarify instructions for updating product formats • Updated Section 7, simplifying product naming convention to remove product and processor version numbers, and processing station ID. Swapped order of processing mode and processing time fields in order to improve readability • Updated the Excel tables Annex to match the above changes to the main text • Corrected size of INST field in product name from 3 to 4 characters (consistent with relevant field in MPHR) • Removed record numbers from Generic Record Header • Modified definition of IPR to allow direct access to records rather than sections within the product • Refined ordering rules for IPRs and fields within IPRs • Defined IPR subclasses • Removed the redundant concept of Associated Data Records (ADRs) throughout document. All relevant data will be placed in the Measurement Data Record. • Section 5.5.1 GRAS MDR is per occultation, not sample interval • Removed SUBSET_TIME and SUBSET_FLAG from the product name as they are redundant. • Add flag to MPHR to indicate when UMARF GIADR is present in a product • Confirm UMARF as a value for INSTRUMENT_GROUP, but renamed to ARCHIVE for more general purpose use • Remove SATELLITE as an INSTRUMENT_GROUP value • Add SEM, ADCS and SBUV as INSTRUMENT_GROUP values • Removed POS from list of enumerated PROD_TYPE • Renamed FUL to SZF in list of enumerated PROD_TYPE for compatability with existing SZO and SZR types. • Removed TBC on two’s complement coding for integers

			• Add DUMMY instrument group type for DMDRs • Nominal NRT Processing mode is given enumerated value of N (originally was O for Operational) in PROC_MODE definitions • Removed all TBDs and TBCs • Introduced new compound type – Variable Scale Factor Integer – to cope with values which cannot be assigned scale factor at this stage. • Updated Level 0 records to correspond to latest Level 0 processing tasks (i.e. unpacking of VCDUs and related error conditions) • Updated DISPOSITION MODE definitions to be mutually exclusive. Removed Validation. Added Testing. • Updated MPHR field names to give them longer, more meaningful names for display in ASCII format • Updated MPHR to have 4 parent product names. Accounts for cases like ATOVS Level 2 (up to 4 input products) and IASI (where data from other products is used). • Corrected definition of UTC time component DD from “day of year (1 – 366)” to “day of month (1 –31)”
6.1	27/03/02		• Updated product name to include DISPOSITION_MODE • Corrected size of product name to 67 characters • GENE deleted from list of INSTRUMENT_ID options • Updated definition of IPR record and added details on its use. Reverted POINTER compound type definition back to Issue 5 Rev 2 version • Removed RECORD_STATUS from GRH. • Added MISSING_DATA_FLAG to DMDR to indicate reason for data missing. Updated and clarified enumerated value settings for reasons for missing data loss • Corrected INSTRUMENT_GROUP setting for NOAA L0 MDRs from ATOVS to NOAA (as MDR will contain more than just NOAA instruments). • Added record subclass version numbers to GRH description tables for each record • Added product major/minor version numbers to generic L0 product description section • Updated description of VEADR and GEADR. Content is recognised by the RECORD_SUBCLASS field (not just by the auxiliary data name). • SOURCE_DATA_TYPE field deleted (was present in text but not in MPHR definition) • Changed INSTRUMENT_ID value AMSU to AMSA to be compatible with CGSRD
6.2	27/05/02	DRAFT	• Changes made due to mini-RIDs raised upon the document and comments received from Alcatel • Reverted to Issue 5.2 definition of an IPR per GAD, VAD and body section only. Number of IPRs is then 3, not 5. RECORD_SUBCLASS of the IPR reverted back to a value of 0 (as in issue 5.2) • Updated Figure 2 to reflect the 3 IPRs for the GAD,VAD and Body sections.

			• Updated use of Major and Minor Format Version numbers. • Section 8 Product Format Version Control updated to correspond to the new usage of Major/Minor Format version fields. • Redundant definition for SRC_DATA_TYPE field deleted • Updated filed names that were not in line with current names in the Annex • Removed “Others (TBD)” entry from INSTRUMENT_ID field enumerated values • Removed “Others (TBD)” entry from PRODUCT_TYPE field enumerated values • Removed “Others (TBD)” entry from PRODCESSING_CENTRE field enumerated values • Removed “Others (TBD)” entry from SOURCE_GROUND_STATION field enumerated values • Removed “Others (TBD)” entry from INSTRUMENT_MODEL field enumerated values • New level 0 product created for VCDU34 data. Identified by INSTRUMENT_ID value of HKTM. Value added to table. • Added AIP and TIP to PRODUCT_TYPE values to account for SAIP and STIP data. • Corrected test in 5.4.2, replacing VEADR/GEADR with VIADR/GIADR. • Corrected text in 5.4. Contents9 of a VIADR are VAD data (corrected from GAD data). • Added information about presence of GIADRs with INSTRUMENT_GROUP = ARCHIVE in 4.2.1.1.2 • Renamed MPHR field SOURCE_GROUND_STATION to RECEIVING_GROUND_STATION to help clarify the functionality of this field. Updated description to provide clarification. • MISSING_DATA_FLAG within DMDR renamed to LOST_DATA_FLAG to be consistent with above changes
6.2	12/06/02	EUM.EPS.SYS.DCR.02.130	• Simplified ASCII record formats by removing the equivalent binary record concept. Now simple generic types, INTEGER, BOOLEAN, etc. are used to indicate the type of data that is encoded in the ASCII field. • Added info on Level 0 VIADR containing UTC-OBT conversion parameters • Updated Section 8 on product format version control. Version numbering to be based on incremental changes and not document version numbers. Record versions and product format versions to be recorded in GPFS. Added initial table of all products and all records. • Removed RECORD_SUBCLASS_VERSION from all GRH definition tables. These will now be defined in one place in Section 8. • Added clarification of processor version numbering

			• Added clarification of use of GEADRs and VEADRs. Updated definition of pointer within GEADRs/VEADRs. • Clarified that integer data types are Big-Endian • Added long general time data format to allow millisecond accuracy when quoting times within ASCII format records [RID answer].
6.3	09/09/02	EUM.EPS.SYS.DCR.02.143	For full details of changes, see DCR • Amended Section 5.2.1 for use of IPRs • Deleted extraneous text at end of Section 9.5.1 • Table 27: Added Validation as a processing mode in line with CGSRD • Section 2.2.1.2: Added text for clarification of bit ordering • Section 2.2.1.3: Added text for clarification of new line character • Corrected acronym GRH where it was wrongly typed as GHR throughout document • Section 4.1: Updated text for clarification • Section 4.2.1.1.6: Added text to explain time-tagging of MDRs for NOAA L0 data • Amend text in Section 5 • Add HRPT and LRPT values to Table 23 • Amend applicable products for GAC in Table 23 • Add RUSx value to Table 26 • Add text to Section 5.1.1.2.9 • Add text to Section 5.5.2.1 • Add text to Section 5.5.2 • Amended Table 51 • Added values to Table 52 • Added text to Section 6 • Added Section 5.1.1.2.11 • Added Section 5.1.1.2.12 • Added Section 5.1.1.2.13 • Major update to Section 5.5.2.2.1 and Table 43. • Major update to Section 9.3. Section 9.3.1 removed. Table 49 removed. • Deleted text in Section 5.1.1.2.8 • Added RUS value to Table 29 • Update of Figure 2 • Added Section 2.4. Added Table 13 • Updates made to Annex (these are described in the DCR at front of Annex) • Deleted Table 11 and Table 12. Redundant with Annex. Added text to Section 2.3.3.1 and 2.3.3.2. • Added ANNEX A1 page to appear in table of contents • Regenerated all Section, Table and Figure numbering. • Regenerated Table of Contents, List of Tables and List of Figures • Updated headers with new issue data • Added Section 5.6 and Table 45 for generic record format version control • Added Section 9.6 and Table 53 for L0 record format

			version control • Added reference document RD-223 and updated text in Section 8. Added reference to this document for product format version control. Deleted Table 47 (Product Format Version Control for all EPS products) • Deleted Section 9.2 and Table 48 (Level 0 product Version)
6.4	10/02/03	EUM.EPS.SYS.DCR.03.011	• Changed text in Section 5.2.1 to provide correction on use of IPRs • Deleted text in Section 4.2.1.1.4 to remove requirement on updating records • Corrected levels of sub-headings in Section 5.3.1 • Added text to Section 5.3.1 to provide clarification on when GEADRs are generated • Added text to Section 5.4.1 to provide clarification on when VEADRs are generated • Changed text in Section 9.4 to provide details of NOAA and METOP L0 MDR contents • Added IASI_L2 instrument group to Section 4.2.1.1.2, Table 14 and clarified use of IASI entry • Updated signature table
6.5	16/02/05	EUM.EPS.SYS.DCR.04.0082 EUM.EPS.SYS.DCR.05.0192	• Update to VIADR-L0-OBT2UTC in Annex • Clarification of reference frames for the orbit state vectors in the MPHR. Added Section 5.1.1.2.14. Added relevant frame type to descriptions of OSV fields in MPHR in Annex. • Regenerated Tables and Numbering for Contents, Tables and Figures
6.6	12/09/05	EUM.EPS.SYS.DCR.05.0254	• Update table 29 in Section 5.1.1.2.10 to allow for further flight model versions
v7A	20/08/08		Migrated into Hummingbird. Body contents copied into standard template. Editorial updates only: - Signature table updated. - Document references automated using bookmarks. - Corrections for typos and spelling standards, and formatting tidied up.
v7B	29/01/09		• Inclusion of all updates representing successive draft versions of 6.7 (up to Draft 5), originally made to document before its transfer to Hummingbird. Details below. • Annex 1 retitled Appendix A, and link added to Annex file in Hummingbird.
	12/03/07	EUM.SYS.DCR.07.0343	• Response to AR.4603 and AR.3502 • Added text to Section 4.2.1.1.6 to indicate that the 1ms resolution of the record start and stop times may cause 1ms overlaps or gaps between consecutive MDRs.
	12/03/07	EUM.SYS.DCR.07.0348	• Response to AR.6004 • Added text to section 9.3 to indicate that VIADR OBT2UTC is a MetOp record that can occurs 1 or

			more times in nominal situations, or zero times in non-nominal cases
	12/03/07	EUM.SYS.DCR.05.0255	• Response to AR 5595 • Correct scaling factor for EARTH_SUN_DISTANCE_RATIO field in MPHR • Add referenced value for 1 AU required to calculate this ratio
	12/03/07	EUM.SYS.DCR.05.0290	• Response to NCR 1508 • Table 32, change IPR record subclass to zero • ANNEX: IPR > POINTER > Change to description
	12/03/07	EUM.SYS.DCR.06.0341	• Response to AR 6822 and AR 6821 • Update Section 5.5.2 • Update Section 5.5.2.2.1 • Update Table 42 • Update ANNEX > Dummy MDR
	19/04/07	EUM.SYS.DCR.07.0358	• Clarification added to descriptions for ANNEX> MPHR> ORBIT_START and ORBIT_END
	19/04/07	EUM.SYS.DCR.07.0359	• Response to AR 7682 • Extended INSTRUMENT_MODEL enumerated values to cover case of multiple instruments or not applicable.
	26/07/07	EPS_AB_DCR_EUM_1	• Response to AR 8184 • Extend table 25 to add new Processing Centres • Update Document Signature Table
	01/10/07	EPS_AB_DCR_EUM_11	• Response to EPS_AB_ECP_135 • Added Section 9.4.2 – Use of DEGRADED_INST_MDR flag in L0 MDRs.
	02/11/07	EUM.SYS.DCR.06.0336	• Response to EUM.EPS.AR.1675.6 • Added note to Table 15 and Section 9.4.1 about record stop time set to record start time for L0 MDRs.
	26/02/08		• Correction of minor typos; also corrected previous occurrences of 'INST_DEGRADED_ MDR' and 'PROC_DEGRADED_ MDR' flags to 'DEGRADED_INST_MDR' and 'DEGRADED_PROC_MDR' respectively (Section 5.1.1.2.11).
	16/05/08	EPS_AB_ACTION_6417	• Response to EPS_AB_ECP_301 • Added new DISPOSITION_MODE value 'E' to Table 27. • Added EARS to Acronyms list.
	29/01/09	ODT_DCR_3	• Section 8 (Product Format Version Control) rewritten for better clarification.
v7C	20/10/09	ODT_DCR_100 (after EUM.EPS.AR.8576)	• Section 5.1.1.2.6 text updated.
		ODT_DCR_101	• Table 48 record name 'VIADR-UTC-2-OBT' corrected to 'VIADR-L0-OBT2UTC' to agree with name in Annex worksheet.
			• Other minor editorial edits, including replacing EUMETSAT logo in page headers, and enhancing App. A intro text as for other PFS documents.
v7D	17/11/09	ODT_DCR_104	• Section 5.1.2.1 Table 30 updated.
v7E	12/03/10	ODT_DCR_134	• Sections 2.2.1.1 & 2.3.1.1: Clarification of application of scale factor.

v7E (cont.)	30/11/10	ODT_DCR_220	• Clarification that, for subset product, the file name and PRODUCT_NAME field of the MPHR are to reflect the sensing start/end times of actual subset. - Section 5.1.1.2.13: Sentence added. - Annex MPHR: Update to descriptions of SENSING_START and SENSING_END fields.

Table of Contents

List of Tables

List of Figures

1 INTRODUCTION

1.1 Purpose and Scope of Document

This document is the Generic Product Format Specification for the products to be produced by the Product Generation functions in the EUMETSAT Polar System (EPS) Core Ground Segment (CGS).

The purpose of this document is to specify the generic product specification for all EPS products, to identify and detail elements that are common to all products, and to define the generic Level 0 product format.

This document is applicable for all Level 0, Level 1 and Level 2 products to be produced by the EPS-CGS. The EPS product processing levels are defined in [**AD-49**].

This document is complemented by the EPS Product Format Specification documents for each of the Instrument Chains in the EPS System.

1.2 Structure of the Document

The document is organised in the following sections:

Section 1	Introduction and description of the scope of the document
Section 2	Definitions of data types used to encode information in an EPS product
Section 3	Definitions of fields within an EPS product
Section 4	Definitions of records within an EPS product
Section 5	Definitions of section within an EPS product and detailed formats for the various record classes that exist within a specific section
Section 6	Final overview of EPS products, regional products and PDUs
Section 7	The naming convention for EPS products
Section 8	The Format Version Control for EPS products
Section 9	Description of the Generic EPS Level 0 product
Appendix A	Link to Annex providing record format descriptions of generic records and records specific to Level 0 products

1.3 Conventions

This document follows the conventions specified in [**AD-48**].

All acronyms found within this document and not specifically described herein are specified in the glossaries of [**AD-49**] and [**AD-48**].

Where a deviation from or limitation to the use of the convention is required it will be explicitly defined within this document. In particular, in the case of a conflict between definitions in other documents and this document, the conventions in this document shall take precedence.

1.4 Terminology

The following terms and definitions are used within this document:

Product	A product is produced by a Product Generation Function. EPS products are listed in [**AD-49**].
Section	A section is a conceptual grouping of related records within a product. It has no physical identity within the product.
Record	A record is a collection of related fields with an identifying header.
Field	A field contains a data type or an array of data types.
Data Type	A data type is a way to encode information in a product.

1.5 Acronyms

AD	Applicable Document
ADR	Associated Data Record (Obsolete)
CCSDS	Consultative Committee for Space Data Systems
CGSRD	Core Ground Segment Requirements Document
DMDR	Dummy Measurement Data Record
EARS	EUMETSAT Advanced Retransmission Service
GAD	Global Auxiliary Data
GEADR	Global External Auxiliary Data Record
GIADR	Global Internal Auxiliary Data Record
GRH	Generic Record Header
IPR	Internal Pointer Record
MDR	Measurement Data Record
MPHR	Main Product Header Record
NRT	Near Real Time
PCD	Product Conventions Document
PDU	Product Dissemination Unit
PGF	Product Generation Function
PFS	Product Format Specification
RD	Reference Document
SPHR	Secondary Product Header Record
VAD	Variable Auxiliary Data
VEADR	Variable External Auxiliary Data Record
VIADR	Variable Internal Auxiliary Data Record

1.6 Applicable Documents

AD-48	EPS Mission Conventions Document, EPS/SYS/SPE/990002.
AD-49	Core Ground Segment Requirements Document, EPS/GGS/REQ/95327

1.7 Reference Documents

RD-1	MetOp Space to Ground Interface Specification, MO-IF-MMT-SY0001, Issue 5.0, 17 October 2000
RD-107	NOAA N, N' Space to Ground Interface, 06887-IS23033284, Issue 1.0, 14 August 2000
RD-136	EUMETSAT Polar System (EPS) Design Justification Document Appendix A4: Mission Analysis, EPS.SYS.TEN.990031, Issue: 1.0, 17 December 1999
RD-223	EPS Product Format Version Control, EUM.EPS.SYS.TEN.02.014

2 DATA TYPES

2.1 Introduction

A data type describes how information is encoded in an EPS product.

The data types used for an EPS product may be divided into two sub-types: basic data types and compound data types.

Any data type may also be present as an array in the product. A one-dimensional array, **X**, runs from $\mathbf{X_0}$ to $\mathbf{X_{n-1}}$ where **n** is the number of elements in the array. An array, **X**, of more than one dimension, e.g. **X(i, j, …, p)**, will be stored in column-major order where the first subscript, **i**, varies most rapidly, the second, **j**, varies next most rapidly, and so on to the last subscript, **p**, which varies least rapidly.

2.2 Basic Data Types

Any EPS product may only use the basic data types that are defined within this document. No PFS document may define additional basic data types.

2.2.1 Integers

There are four basic types of integer.

Signed integers use the "Two's Complement" coding convention for negative values. They can be 1, 2, 4, and 8 bytes in size, and have type identifiers of byte, integer2, integer4 and integer8 respectively. They can hold values in the range from -2^{n-1} to $+2^{n-1}-1$, where **n** is the length of the integer in bits (8 bits per byte).

Unsigned integer types are positive integers. They are also 1, 2, 4, and 8 bytes in size and have type identifiers u-byte, u-integer2, u-integer4 and u-integer8, respectively. They can hold values in the range from 0 to $+2^{n}-1$ where **n** is the length of the integer in bits.

Enumerated integers may only contain a value from a set of specified integer values, each of which is associated with a named concept, e.g. a set of error codes. When this field type is defined, the possible integer values and associated names shall be specified. For EPS products, an enumerated integer is always one byte in size (allowing 256 distinct "flag" states) and will be indicated by the special type identifier, enumerated.

Boolean integers are a specific enumerated integer type which takes only 2 possible values: when all bits are zeroed, it denotes 'FALSE', otherwise, if any bit is set (i.e. its value is different from zero), it denotes 'TRUE'. For EPS products, a Boolean will always be one byte in size and will be indicated by the special type identifier, boolean. (For efficient coding of Boolean information, see also the definition for bit strings.)

These Integer types are summarised by size in Table 1.

All integers will be in Big-Endian format i.e. most significant byte to least significant byte order.

Type ID	Type	Bytes	Range
byte	Signed Byte	1	-128…127
u-byte	Unsigned Byte	1	0…255
enumerated	Enumerated Byte	1	256 flag states
boolean	Boolean Byte	1	False/True
integer2	Signed 2-byte Integer	2	-32768…32767
u-integer2	Unsigned 2-byte Integer	2	0…65535
integer4	Signed 4-byte Integer	4	-2147483648…2147483647
u-integer4	Unsigned 4-byte Integer	4	0…4294967295
integer8	Signed 8-byte Integer	8	-9223372036854775808… 9223372036854775807
u-integer8	Unsigned 8-byte Integer	8	0… 18446744073709551615

Table 1: Summary of Integer Types (organised by size)

2.2.1.1 Reals

There will be no “real” data types within a product. This is to avoid problems encountered when moving between computer architectures. Instead, all real values will be encoded as integer values by employing a power of ten multiplier - the scale factor - dependent upon the accuracy with which the value needs to be stored. This scale factor will be specified within the tables that describe the record format. The real value is calculated by dividing the integer value by the scale factor expressed as a power of ten. For example, if the integer value = 123456 and the scale factor = 8, then:

real value = $123456 \times 10^{-8} = 0.00123456$

2.2.1.2 Bit Strings

A bit string is encoded as follows:

$\mathbf{b_{n-1}...b_0}$

where $\mathbf{b_i}$ is the i^{th} bit in the string and **n** is the length in bits of the bit string, with $\mathbf{b_{n-1}}$ being the most significant bit. The value of **n** shall always be a multiple of 8 ensuring that a bit string is always a full number of bytes in size.

Any individual bit in the bit string may be set to 0 or 1, and so the bit string can function as a method of compressing a set of Boolean fields into a more compact entity.

Type	Type ID	Size	Notes
Bit String	bitst(**n**)	1 bit per element	

Table 2: Bit String Field Type

2.2.1.3 Character Strings

A character string is a sequence of one or more 8-bit ASCII characters, each character occupying one byte, as defined in the ANSI X3.4 norm.

There are three types of character strings within EPS products.

The first type is the standard character string (char), which can contain only upper cased letters [A…Z], numbers [0…9] and the underscore character "_". The standard character string has a type identifier, char, and the number of characters in the string is specified by a **length** parameter, e.g.

char(**length**)

A 32 character string will therefore be represented as char(32).

The second type is the enumerated character string, with a type identifier of e-char(**length**). This can only contain one of a set of specified string values (e.g. one of the 4-letter codes for designating the Metop instruments). These specified string values must be of the same length as the enumerated character string. The enumerated character string obeys the content restrictions on the standard character string except that it may also contain lower case "x" characters. These are used in place of white space to pad specified string values to the same length as the enumerated character string.

The final type is the extended character string, with a type identifier of x-char(**length**). This shares the same characteristics as the standard character string, except the allowed contents are extended to include the space character, the new line character (line feed without carriage return - represented as **n**), the equals sign (=) and the plus (+) and minus (-) signs. Extended character strings are found only in ASCII records (see Section 4.3).

Type	Type ID	Size	Notes
Standard Character String	char(**length**)	1 byte per character	Can only contain upper case letters [A…Z], numbers [0…9] and the underscore character (_). The number of characters in a character string is determined by the **length** parameter e.g. CHAR(8) is an 8 character string.

Enumerated Character String	e-char(**length**)	1 byte per character	Same properties as standard character string except that it can only contain one of a set of specified string values, and may also include the lower case “x” character (used as whitespace padding).
Extended Character String	x-char(**length**)	1 byte per character	Same properties as standard character string except that it may also contain space character, the newline character (\n), the equals sign (=) and the plus (+) and minus (-) signs. Only found in ASCII records.

Table 3: Character String Data Types

2.2.2 Time Formats

2.2.2.1 Generalised Time

Standard generalised time or generalised time is UTC based. It is a string of 15 characters with the terminator character of an ASCII “Z” (indicates Zulu or UTC time). The format is:

YYYYMMDDHHMMSSZ

where:

YYYY stands for the year (e.g. 2005)
MM for the month of the year (1 to 12)
DD for the day of the month (1 to 31)
HH for the hours in the day (00 to 23)
MM for the minutes in the hour (00 to 59)
SS for the seconds in the minute (00 to 59)
Z is always the last character

For example 12:53:22 on 23 March 2005, would be encoded as:

20050323125322Z

Type	**Type ID**	**Size**	**Notes**
Generalised Time	general time	15 bytes	This is a char(15) data type with a specific format

Table 4: Generalised Time Field Type

Generalised time is accurate only to seconds and should be used only where the time needs to be in human-readable format e.g. in fields which are present in the product name. For greater accuracy and compactness within the product, short or long CDS time types should be employed (see later), or, for an ASCII record, long generalised time.

If a field has a type of general time, but no time is applicable then the field should be filled with the string for "no applicable time", which is a string of 14 lower case 'x' characters terminated by the ASCII character 'Z', e.g.

xxxxxxxxxxxxxxZ

2.2.2.2 Long Generalised Time

Long generalised time is UTC based and has similar characteristics to the generalised time data type. It is a string of 18 characters with the terminator character of an ASCII "Z" (indicates Zulu or UTC time). The format is:

YYYYMMDDHHMMSSmmmZ

where:

YYYY stands for the year (e.g. 2005)
MM for the month of the year (1 to 12)
DD for the day of the month (1 to 31)
HH for the hours in the day (00 to 23)
MM for the minutes in the hour (00 to 59)
SS for the seconds in the minute (00 to 59)
mmm for the milliseconds in the second (000 to 999)
Z is always the last character

For example 12:53:22.458 on 23 March 2005, would be encoded as:

20050323125322458Z

Type	**Type ID**	**Size**	**Notes**
Long Generalised Time	long general time	18 bytes	This is a char(18) data type with a specific format

Table 5: Long Generalised Time Field Type

Generalised time is accurate only to seconds and should be used only where the time needs to be in human-readable format e.g. in fields which are present in the product name. For greater accuracy and compactness within the product, short or long CDS time types should be employed (see later).

If a field has a type of long general time, but no time is applicable then the field should be filled with the string for "no applicable time", which is a string of 17 lower case 'x' characters terminated by the ASCII character 'Z', e.g.

xxxxxxxxxxxxxxxxxZ

2.3 Compound Data Types

A compound data type is composed of at least two basic and/or other compound data types (including arrays of data types). As well as the generic compound data types described in this document (below) which may be used by all EPS products, there may also be compound data types that are unique to a particular product and which will be detailed in the PFS document for that product. These PFS-specific compound data types may only be constructed from basic data types that are defined within this document (i.e. a PFS may not define new basic data types, only new compound data types). Within the PFS document, the type ID of the compound data type is written in upper case in order to aid identification within the record format description tables.

The compound data types are given a name that is unique to that particular PFS document and that does not clash with any data type names within this document.

One advantage of using a compound data type is that when a compound data type field is specified as an array, the related information that comprises the compound data type is kept grouped together within the array.

2.3.1 Integer Formats

2.3.1.1 Variable Scale Factor Integers

The EPS product format specification does not allow "real" data types to be present in a product. Instead, real values are encoded into integer format using a fixed scaling factor that is specified in the format specification tables. However, there may be some values that vary too much to be efficiently encoded into an integer value with a fixed scaling factor. If these are single values, they may be encoded into a compound that includes a scaling factor and the integer value as described in this section. If these values are an array of values, they are more easily presented by an array of bytes containing the variable scale factors followed by an array of integer data types. Section 2.2.1.1 describes how to apply a scale factor to calculate the required real value.

Type	Type ID	Size (bytes)	Components
Variable Scale Factor Byte	V-BYTE	2	byte + byte
Variable Scale Factor Unsigned Byte	VU-BYTE	2	byte + u-byte
Variable Scale Factor Integer-2	V-INTEGER2	3	byte + integer2
Variable Scale Factor Unsigned	VU-INTEGER2	3	byte + u-integer2

Integer-2			
Variable Scale Factor Integer-4	V-INTEGER4	5	byte + integer4
Variable Scale Factor Unsigned Integer-4	VU-INTEGER4	5	byte + u-integer4
Variable Scale Factor Integer-8	V-INTEGER8	9	byte + integer8
Variable Scale Factor Unsigned Integer-8	VU-INTEGER8	9	byte + u-integer8

Table 6: Variable Scale Factor Integers (byte component contains scale factor)

2.3.2 Time Formats

2.3.2.1 Short CDS Time

The Short CCSDS Day Segmented (CDS) time encodes the day since epoch in the first 2 bytes and the number of milliseconds since the beginning of the day in the day in its last 4 bytes as below:

Day	Milliseconds of day
u-integer2	u-integer4
2 bytes	4 bytes

Table 7: Short CDS Time Components

The epoch time is 1 January 2000 starting with 0. The CDS time is UTC-based and takes into account leap second corrections.

Type	Type ID	Size	Components
Short CDS Time	short cds time	6 bytes	u-integer2 + u-integer4

Table 8: Short CDS Time Field Type

2.3.2.2 Long CDS Time

The Long CCSDS Day Segmented (CDS) time encodes the day since epoch in the first 2 bytes, the number of milliseconds since the beginning of the day in its next 4 bytes, and the number of microseconds since the last millisecond in its last 2 bytes.

Day	Milliseconds of day	Microseconds of millisecond
u-integer2	u-integer4	u-integer2
2 bytes	4 bytes	2 bytes

Table 9: Long CDS Time Components

The epoch time is 1 January 2000 starting with 0. The CDS time is UTC-based and takes into account leap second corrections.

Type	Type ID	Size	Components
Long CDS Time	long cds time	8 bytes	u-integer2 + u-integer4 + u-integer2

Table 10: Long CDS Time Field Type

2.3.3 Other Generic Compound Data Types

2.3.3.1 REC_HEAD

Every record within an EPS product commences with a generic record header which consists of a compound data type REC_HEAD. This record header contains metadata about the type and size of the record. This is described in detail in Section 4.2 and in the Annex (Appendix A).

2.3.3.2 POINTER

The POINTER field type is used as an internal record pointer within the EPS product. It provides direct access to a record. It is used solely within Internal Pointer Records as described in Section 5.2.1 and its structure is detailed in the Annex.

2.4 Undefined Values for Data Types

Default values for fields will be defined in the relevant PFS document. However, there may be cases were an “undefined” value needs to be inserted into a field to flag undefined values (similar to the “NaN” or “Not a Number” values used for IEEE real data types). An example may be where data is corrupted and is therefore too large to fit into the assigned field within a product.

Booleans and bit strings should never have an undefined state and enumerated data types should have “undefined” values defined in the relevant PFS if it is seen necessary to have a value for “undefined”.

Unless otherwise stated, compound data types should be treated by the rules that apply to their component data types.

Any PFS can over-ride these default undefined values if necessary to something more suitable on a field-by-field basis.

Type	Undefined Value
Boolean	N/A
Enumerated types	N/A
Integer Type	Minimum negative value allowed for size of integer
U-Integer Type	Maximum positive value allowed for size of u-integer
Bit String	N/A
Char String	Lower case “x” in place of each character

Table 11: Pointer Field Type

3 FIELDS

The most basic information component of a product is a “field”. The information in a field may be encoded either in a single data type, or an array of data types. Only data types found within this document, or compound data types that are defined locally in the relevant PFS document, may be used.

3.1 Field Names

Each field is identified by a meaningful identifying name, the field name, that is unique to the record within which the field occurs.

For binary records (Section 4.3), the name is purely an identifying tag to help with identification of a field. However, for ASCII records (Section 4.3), the field name is used to create part of the field itself and follows the rules for the standard char string, char(), e.g. it is composed only of upper case letters [A…Z], numbers [0…9], and underscore characters, _, the latter to help with legibility if necessary.

4 RECORDS

A record consists of a generic record header (GRH) which contains metadata about the record, and a contiguous group of related fields holding the information content for the record.

4.1 Record Classes and Subclasses

Each record belongs to one of the eight classes as identified in the generic record header (see Section 4.2). Each class of record may be assigned to a particular instrument group. This instrument group may in turn define subclasses of the record class as required.

Example: Thus there is one class of Measurement Data Record (MDR). This class exists for a number of instrument groups (e.g. IASI, HIRS, GOME, etc.). In addition, there may be a number of subclasses of a particular instrument group MDR in order to hold different types of measurement data in an efficient manner. The GRH uniquely identifies a given subclass of record.

Any given class of record is confined to a specific section of a product. Section 5 describes the sections within a product and also details the various record classes found within each product section.

4.2 Generic Record Header (GRH)

Every record in an EPS product has a generic record header (GRH). It contains the metadata necessary to uniquely identify the record class, subclass and occurrence, and also to ensure correct linking and time-ordering of records within the product.

Field	**Description**
RECORD_CLASS	Class of this record
INSTRUMENT_GROUP	Group defining the record subclasses
RECORD_SUBCLASS	Subclass of this record class
RECORD_SUBCLASS_VERSION	Format version of this record subclass
RECORD_SIZE	Total size of the record in bytes (including GRH)
RECORD_START_TIME	Start Time for this record – context will depend on record class
RECORD_STOP_TIME	Stop Time for this record – context will depend on record class

Table 12: Generic record header fields

The detailed format of the GRH is given in the Annex (Appendix A) to this document under the ‘Compounds’ worksheet.

Further details on the fields of the GRH are given below.

4.2.1.1 Generic Record Header Fields

4.2.1.1.1 RECORD_CLASS

For any instance of a record in a product, the record class field identifies the record class of which it is a member. It is an enumerated key, as set out in the table below.

Index	Record Class	Acronym
0	Reserved	
1	Main Product Header Record	MPHR
2	Secondary Product Header Record	SPHR
3	Internal Pointer Record	IPR
4	Global External Auxiliary Data Record	GEADR
5	Global Internal Auxiliary Data Record	GIADR
6	Variable External Auxiliary Data Record	VEADR
7	Variable Internal Auxiliary Data Record	VIADR
8	Measurement Data Record	MDR

Table 13: RECORD_CLASS enumerated values

4.2.1.1.2 INSTRUMENT_GROUP

The Instrument Group field contains an enumerated integer that identifies the group responsible for defining the particular class and subclass (if any) of the record.

Index	Defining Group
0	GENERIC (no specific instrument)
1	AMSU-A
2	ASCAT
3	ATOVS instruments (AVHRR/3, HIRS/4, AMSU-A, MHS)
4	AVHRR/3
5	GOME
6	GRAS
7	HIRS/4
8	IASI (except IASI L2 products)
9	MHS
10	SEM
11	ADCS
12	SBUV
13	DUMMY
14	ARCHIVE (Note: Only used in GIADRs. A GIADR with INSTRUMENT_GROUP of archive contains only descriptive information and is not processed.)
15	IASI_L2 (used for IASI L2 products only)

Table 14: INSTRUMENT_GROUP enumerated values

4.2.1.1.3 RECORD_SUBCLASS

This is determined by the Instrument Group and shall vary from instrument to instrument and also, if necessary, from processing level to processing level. The record subclasses are defined in the relevant PFS document.

4.2.1.1.4 RECORD_SUBCLASS_VERSION

This is the version number of the record subclass. Any update to the format of the record subclass shall result in the increment of the subclass version number.

4.2.1.1.5 RECORD_SIZE

This field contains the total size of the record subclass (including the GRH) in bytes.

4.2.1.1.6 RECORD_START_TIME and RECORD_STOP_TIME

These fields contain time tags for the record that are used to ensure that records are kept in the correct chronological order and to provide linkage between records of various classes. The time tags are based on the time domain used by the GRH of the MDRs. This is usually based on the sensing time of the data, but the individual PFSs should be checked for local definitions. The precise meanings of the time tags will also depend on the record class as shown in Table 15.

Record Class	Record Start Time	Record Stop Time
Main Product Header Record	RECORD_START_TIME of the first MDR in the product	RECORD_STOP_TIME of the last MDR in the product
Secondary Product Header Record	RECORD_START_TIME of the first MDR in the product	RECORD_STOP_TIME of the last MDR in the product
Internal Pointer Record	RECORD_START_TIME of the first MDR in the product	RECORD_STOP_TIME of the last MDR in the product
Global External Auxiliary Data Record	RECORD_START_TIME of the first MDR in the product	RECORD_STOP_TIME of the last MDR in the product
Global Internal Auxiliary Data Record	RECORD_START_TIME of the first MDR in the product	RECORD_STOP_TIME of the last MDR in the product
Variable External Auxiliary Data Record	The RECORD_START_TIME of the first MDR for which this data applies.	The RECORD_STOP_TIME of the last MDR for which this data was applied.
Variable Internal Auxiliary Data Record	The RECORD_START_TIME of the first MDR for which this data applies.	The RECORD_STOP_TIME of the last MDR for which this data was applied.
Measurement Data Record	Usually the “sensing time” of the first measurement in the record, but see individual PFSs for local definitions	Usually the “sensing time” of the last measurement in the record, but set to RECORD_START_TIME for Level 0. See also individual PFSs for local definitions

Table 15: Definitions of Record Start Time and Record Stop Time fields

Within the Generic Record Header of a NOAA L0 MDR, the record start time is set for GAC frames to the time code derived from the GAC frame. For SAIP or STIP frames, record start time is set to the time code derived from the relevant TIP frame. In all cases, the record stop time is derived from the duration of the frame. Details of the time-coding of NOAA frames are available from [**RD-107**].

The sensing start and stop times in the MPHR will be updated as normal, but will necessarily reflect frame times rather than sensing times.

It should also be noted that the millisecond resolution of the times in the record start time and record stop time fields may cause 1 millisecond gaps or over laps between consecutive MDRs due to rounding errors. Neither of these cases should be considered as non-nominal.

The 1 millisecond gap is caused when the stop time of MDR n is rounded down to the nearest millisecond and the start time of MDR n+1 is rounded up to the nearest millisecond.

The 1 millisecond overlap can be caused by the fact that for some MDRs, the record stop time is not measured but is derived from the record start time plus the expected duration of the record. The accumulation of rounding errors can cause this calculated record stop time for MDR n to overlap with the measured record start time of MDR n +1.

4.3 Record Types

The records that exist in an EPS product may also be divided into two types – binary records and ASCII records. An ASCII record is an ASCII encoding of a binary record in order to render it human-readable.

Both record types have a binary GRH, but ASCII records have fields that are composed solely of the extended character (x-char) data type, with no arrays or compound data types allowed. Binary records have fields composed of any data type *except* extended character data types.

Only the MPHR and SPHR (defined later) exist as ASCII record types. All other records are binary record types.

4.3.1 Fields in ASCII Records

ASCII records consist of the standard binary GRH followed by fields of the extended character (x-char) data type.

Each x-char field of an ASCII record follows the same structure:

```
VALUE_NAME^^^^^^^^^^^^^^^^^^^^=^field_value\n
```

The value name is the first 30 characters of the field name. If the field name is less than 30 characters in length, the rest of the value name is right-padded with ASCII spaces (indicated by the ^ in the example above). There then follows an equals sign, a space, the field value, and finally an ASCII new line character (indicated by the **\n** symbol).

The field value is an ASCII encoding of the equivalent field in the equivalent binary record. For example, information that would be of type u-integer2 in a binary record encoding of this field, will be encoded into 5 ASCII characters consisting of the digits [0…9]. The rules for ASCII encoding of various basic data types are expressed in the table below. Required parameters for the encoding (if any) are given in the relevant record format description tables in the PFS documents.

Data Type	Description
CHARACTER, *E-CHARACTER*	A character string type is simply output as the number of ASCII characters specified in the table. If the specified number of ASCII characters is larger than the number of characters in the string, the output is padded with spaces on the left.
ENUMERATED, *INTEGER,* *U-INTEGER*	The number of ASCII characters in the record format table specifies the number of digits (including any necessary minus sign) that should be output. If the specified number of ASCII characters is larger than the number of digits, the output is padded with spaces on the left.
BOOLEAN	A boolean is output as either an ASCII "F" for False (i.e. all bits unset), or an ASCII "T" for True (i.e. any bit set). It is therefore always one ASCII character in size.
BITSTRING	A bit string is output as a string of ASCII zeroes ("0") and ones ("1"), with each ASCII character corresponding to one bit of the bit string. The number of ASCII characters will always be the same length as the number of bits in the bit string.
GENERAL TIME, *LONG GENERAL TIME*	As generalised time is basically a specialised form of a char(15) type (or char(18) for long form), it is output as 15 (or 18 for long form) ASCII characters as described above.

Table 16: ASCII Encoding for ASCII Records

4.4 Record Format Descriptions

This document has an annex of tables that detail the format of all generic record formats including the generic Level 0 record formats (see Appendix A). The PFS documents also have annexes of tables that detail the format of all other records necessary to construct the particular products they describe.

This section sets out the notation used by all tables that describe the EPS product record formats.

4.4.1 ASCII Record Notation

The description of the format ASCII records is not a direct description of the record format. Instead, it is partly based on the binary format version of the record that is encoded to provide the ASCII format version. This allows the table to show the value that is encoded into the x-char field.

ASCII records are described within tables that use the following column headings:

FIELD	DESCRIPTION	SF	UNITS	TYPE	ENCODE CHARS	FIELD SIZE	OFFSET

where:

Column	Description	Notes
Field	The field name. Every field in a record must have a name that is unique to that particular record subclass.	The first 30 characters of this field name are used to create the value name.
Description	Description of the quantity contained in the field	Every field must have a description.
SF	Scale factor to be used under the form 10^{SF} (the value of the Field must be divided by this scale factor. Example: an angle of 35.6212 degrees with a scale factor of 2 is stored as 3562)	If no scale factor is to be used, this column will be empty.
Units	Units of the quantity contained in the field after conversion by the scale factor and specified according to the conventions in [**AD-49**] or [**AD-48**].	If units are not applicable, this column will be empty.
Type	The theoretical data type that would be used to store the information in a binary record.	This will be one of the basic data types described in this document. It describes the information before it is encoded into ASCII for the field.
Encode Chars	The number of ASCII characters required to contain the ASCII-encoded value of the field.	See Table 16 for encoding rules
Field Size	The total size of the field measured in bytes, including the value name, padding spaces, equals sign, value and new-line terminator.	This is also the length of the x-char data type that will actually comprise this field in the final ASCII record.
Offset	The offset of the field from the start of the record, measured in bytes	

Table 17: Notation used in ASCII record format descriptions

4.4.2 Binary Record Notation

Binary records are described within tables that use the following column headings:

FIELD	DESCRIPTION	SF	UNITS	DIM1	DIM2	…DIMN	TYPE	TYPE SIZE	FIELD SIZE	OFFSET

where:

Column	Description	Notes
Field Name	The field name. Every field in a record must have a name that is unique to that particular record subclass.	
Description	Description of the quantity contained in the field	Every field must have a description.
SF	Scale factor to be used under the form 10^{SF} (the value of the Field must be divided by this scale factor. Example: an angle of 35.6212 degrees with a scale factor of 2 is stored as 3562)	If no scale factor is to be used, this column will be empty.
Units	Units of the quantity contained in the field after conversion by the scale factor and specified according to the conventions in [**AD-49**] or [**AD-48**]	If units are not applicable, this column will be empty.
Type	The data type for the field.	This will be one of the data types described in this document
Dim1, Dim2, …, DimN	Dimensions of the field, from 1st Dimension to Nth Dimension	If the field is not an array (i.e. it is a scalar with dimension = 1), all columns will be filled with value of 1
Type Size	The size of the data type measured in bytes	e.g. a field of type integer2 field will have a type size of 2, a field of type char(32) will have a type size of 32
Field Size	The total size of the field measured in bytes	If the field is a scalar, this is equal to the type size If the field is an array, this is equal to the type size multiplied by the number of array elements
Offset	The offset of the field from the start of the record, measured in bytes	

Table 18: Notation used in binary record format descriptions

5 SECTIONS

A section is a conceptual grouping of related “records” within a product. Unlike a record, it has no physical identity within a product.

In an EPS product there are five sections – the Header Section, the Pointer Section, the Global Auxiliary Data Section, the Variable Auxiliary Data Section and the Body Section. They always occur in this order within the product.

The Header Section contains metadata information that is applicable to the entire product.

The Pointer Section contains pointer information to navigate within the product.

The two auxiliary data sections contain information on the auxiliary data that has been used or produced during the processing of the product. This may either be the actual auxiliary data itself (internal auxiliary data) or a pointer to the source of the data used (external auxiliary data).

The Body Section is usually the main bulk of the product and will contain the instrument measurements or processed instrument measurements and associated information.

Figure 1 shows a brief summary of the sections within an EPS product.

EPS Product
Header Section
Pointer Section
Global Auxiliary Data Section
Variable Auxiliary Data Section
Body Section

Figure 1: Sections within an EPS product

Each product section may contain only certain classes of record, with no record class being used by more than one record section. The five sections and the records specific to each section are described below.

5.1 Header Section

The Header Section contains the metadata for the product and consists of one Main Product Header Record (MPHR), followed by none or one Secondary Product Header Record (SPHR). This is the only section to contain ASCII type records. All other sections contain only binary type records.

Header Section
Main Product Header Record
Secondary Product Header Record

Table 19: Breakdown of Header Section

5.1.1 Main Product Header Record (MPHR)

The Main Product Header Record is common to all instruments and processing levels. It gives information concerning the identification of the dataset and of the spacecraft, observation date, orbit number, etc. It also contains information concerning the processing chain (version of the processing software, date of processing, processing software).

The MPHR is always the first record in any product and is an ASCII record type.

In addition, the product name is formed from the contents of a number of the fields within the MPHR**.**

The Main Product Header Record is detailed in the Annex (Appendix A) to this document.

5.1.1.1 Generic Record Header Fields

The MPHR has the following GRH fields:

Field	Value
RECORD_CLASS	MPHR
INSTRUMENT_GROUP	GENERIC
RECORD_SUBCLASS	0

Table 20: Generic Record Header fields for MPHR

5.1.1.2 Main Product Header Record Fields

The following subsections provide more detailed description of some of the MPHR fields as defined in the Annex.

5.1.1.2.1 INSTRUMENT_ID

Value	Meaning
AMSA	AMSU-A
ASCA	ASCAT
ATOV	ATOVS instruments (AVHRR/3, HIRS/4, AMSU-A, MHS)
AVHR	AVHRR/3
GOME	GOME
GRAS	GRAS

HIRS	HIRS/4
IASI	IASI
MHSx	MHS
NOAA	All NOAA instruments (Specific to Level 0 NOAA product)
SEMx	SEM
ADCS	ADCS
SBUV	SBUV
xxxx	No specific instrument
HKTM	VCDU34 data (Specific to Level 0)

Table 21: INSTRUMENT_ID field enumerated values

5.1.1.2.2 PRODUCT_TYPE

Usage of the PRODUCT_TYPE field is defined by the table of EPS products in Annex B of [**AD-49**].

Value	Meaning	Applicable Products
ENG	IASI engineering data	IASI Level 1 Engineering product
GAC	NOAC Global Area Coverage AVHRR data	NOAA ATOVS Level 0 and AVHRR Level 1 products
SND	Sounding Data	Metop and NOAA ATOVS Level 2 products and IASI Level 2 products
SZF	ASCAT calibrated σ_0 data at full resolution	ASCAT Level 1b
SZO	ASCAT calibrated σ_0 data at operational resolution (50 km)	ASCAT Level 1b
SZR	ASCAT calibrated σ_0 data at research resolution (25 km)	ASCAT Level 1b
VER	IASI verification data	IASI Level 1 Verification product
xxx	No specific product type specified	All other products
AIP	NOAA AIP/SAIP data	NOAA Level 0 products
TIP	NOAA TIP/STIP data	NOAA Level 0 products
HRP	HRPT data	RUS products
LRP	LRPT data	RUS products

Table 22: PRODUCT_TYPE field enumerated values

5.1.1.2.3 PROCESSING_LEVEL

Value	Meaning
00	Level 0
01	Level 1
1A	Level 1a
1B	Level 1b
1C	Level 1c

02	Level 2
03	Level 3
xx	No Specific Level

Table 23: PROCESSING_LEVEL field enumerated values

5.1.1.2.4 SPACECRAFT_ID

Value	Meaning
xxx	No specific spacecraft
Mnn	METOP nn, where nn=01, 02, or 03
Nnn	nn=15, 16, 17, 18, 19 for NOAA-K, L (TBC), M (TBC), N (TBC), N'(TBC)

Table 24: SPACECRAFT_ID field enumerated values

5.1.1.2.5 PROCESSING_CENTRE

Value	Meaning
CGSn	EUMETSAT EPS Core Ground Segment (where n = 1, 2, 3)
NSSx	NOAA/NESDIS
RUSx	Reference User Station
ERFn	EPS Reprocessing Facility (where n = 1, 2, 3)
EARS	EUMETSAT Advanced Retransmission Service
TCEn	Technical Computing Environment (where n = 1, 2, 3)
The following identification of EUM SAF is (TBC) e.g. SAF Host Institutes	
DMIx	DMI, Copenhagen (GRAS SAF)
DWDx	DWD, Offenbach (Climate SAF)
FMIx	FMI , Helsinki (Ozone SAF)
IMPx	IMP, Lisbon (Land SAF)
INMx	INM, Madrid (NCW SAF)
MFxx	MF, Lannion (OSI SAF)
UKMO	UKMO, Bracknell (NWP SAF)

Table 25: PROCESSING_CENTRE field enumerated values

5.1.1.2.6 PROCESSOR_MAJOR_VERSION / PROCESSOR_MINOR_VERSION

The processor version number may be used to uniquely identify all components (software and auxiliary data) that comprise the processor. When not used, the processor version number shall be set to major:1, minor:0. It should be noted that the Level 1 Processor is considered to contain Level 1a, Level 1b and, for IASI, Level 1c processing functionality, so that they share a common processor version number.

All data that do not have a unique reference for use within a GEADR or VEADR are assumed to be under configured under the processor version number.

5.1.1.2.7 PROCESSING_MODE

The EPS system will generate products in the following processing modes as described in the CGSRD **[AD-49]**:

Value	Meaning
N	Nominal (NRT processing)
B	Backlog Processing
R	Reprocessing
V	Validation

Table 26: PROCESSING_MODE field enumerated values

5.1.1.2.8 DISPOSITION_MODE

The EPS system will generate products in the following disposition modes:

Value	Meaning
T	Testing (enables identification of test data)
O	Operational
C	Commissioning
E	Enables identification of data coming from the EARS processing environment

Table 27: DISPOSITION_MODE field enumerated values

5.1.1.2.9 RECEIVING_GROUND_STATION

This field contains the identifying code of the acquisition ground station i.e. the ground station which originally received the down-linked data **[RD-136].** For data received from the NOAA ground segment, this is the predicted ground station (FBK or WAL).

Value	Meaning
SVL	Svalbard
WAL	Wallops Island, Virginia
FBK	Fairbanks, Alaska
SOC	SOCC (NESDIS Satellite Operations Control Centre), Suitland, Maryland
RUS	Reference User Station

Table 28: RECEIVING_GROUND_STATION field enumerated values

5.1.1.2.10 INSTRUMENT_MODEL

The INSTRUMENT_MODEL field identifies the instrument model that has been used to make the measurements.

Value	Meaning
0	Engineering Model
1	Proto-Flight Model / Flight Model 1 (of a series)
2	Flight Model 2 (of a series)
3	Flight Model 3 (of a series)
4	Flight Model 4 (of a series)
5	Flight Model 5 (of a series)
6	Flight Model 6 (of a series)
7	Flight Model 7 (of a series)
8	Flight Model 8 (of a series)
9	Flight Model 9 (of a series)
10	Flight Model 10 (of a series)
11	Flight Model 11 (of a series)
12	Flight Model 12 (of a series)
13	Flight Model 13 (of a series)
14	Flight Model 14 (of a series)
	etc.
255	Not Applicable or Multiple Instruments for this product

Table 29: INSTRUMENT_MODEL field enumerated values

5.1.1.2.11 COUNT_DEGRADED_PROC_MDR and COUNT_DEGRADED_INST_MDR

The MPHR fields COUNT_DEGRADED_PROC_MDR and COUNT_DEGRADED_INST_MDR are the sum across all non-Dummy MDRs in the product of the DEGRADED_INST_MDR and DEGRADED_PROC_MDR fields respectively.

The conditions for the setting of the MDR fields DEGRADED_INST_MDR and DEGRADED_PROC_MDR are described in the relevant PGS and/or PFS document. If no mechanism is currently described, then these fields should be defaulted to Boolean False (0).

5.1.1.2.12 RECEIVE_TIME_START and RECEIVE_TIME_END

For Metop, times are taken from the acquisition stamping of VCDUs.

For NOAA, times are based on the predicted AOS + user-configurable value (zero default) and predicted LOS – user-configurable value (zero default).

5.1.1.2.13 SUBSETTED_PRODUCT

The SUBSETTED_PRODUCT flag is set when the product has been subset during retrieval from the UMARF archive. It indicates the product may vary from the GPFS specification for full products and may also contain one or more GIADRs inserted by the UMARF. These GIADRs are identified by the GRH.INSTRUMENT_GROUP value of ARCHIVE.

Full products retrieved from archive always correspond to the GPFS full product format and will not contain any UMARF GIADRs.

The file name of the subset product, and the corresponding PRODUCT_NAME field of the MPHR, shall reflect the sensing start time and sensing end time of the actual subset.

5.1.1.2.14 State Vector Reference Frames

The Cartesian Orbit State Vector fields will contain the Cartesian Orbit State Vector in the Earth-Fixed reference frame as defined in the EPS Mission Conventions Document [**AD-48**].

The Keplerian Orbit State Vector fields will contain the Keplerian Orbit State Vector in the True-Of-Date reference frame as defined in the EPS Mission Conventions Document [**AD-48**].

5.1.2 Secondary Product Header Record (SPHR)

The Secondary Product Header Record generally contains more detailed, non-generic information at the product level, such as product quality indicators, processing indicators, summed information from MDRs, etc. It may be unique to each instrument and/or product level and thus is defined in the relevant instrument PFS documents.

The SPHR is an ASCII record type.

5.1.2.1 Generic Record Header Fields

The SPHR has the following GRH fields:

Field	Value
RECORD_CLASS	SPHR
INSTRUMENT_GROUP	<see relevant PFS>
RECORD_SUBCLASS	<see relevant PFS>

Table 30: Generic Record Header fields for SPHR

5.2 Pointer Section

The Pointer Section contains Internal Pointer Records (IPRs). These contain pointers to records within the Global Auxiliary Data, Variable Auxiliary Data and Body sections that follow immediately after the Pointer section.

Header Section
Internal Pointer Record(s)

Table 31: Breakdown of Pointer Section

5.2.1 Internal Pointer Record (IPR)

IPRs are generated for the GAD, VAD and Body sections. An IPR is created each time either the Record Class or Instrument Group or Record Subclass in the target record GRH is different from the corresponding field in the previous target record's GRH. The IPRs are ordered by the order of their target records.

The Internal Pointer Record is detailed in the Annex (Appendix A) to this document.

5.2.1.1 Generic Record Header Fields

The IPR has the following GRH fields:

Field	Value
RECORD_CLASS	IPR
INSTRUMENT_GROUP	GENERIC
RECORD_SUBCLASS	0

Table 32: Generic Record Header fields for IPR

5.3 Global Auxiliary Data Section

The Global Auxiliary Data (GAD) is any auxiliary data associated with the product processing that is applicable to the complete product and will not change throughout the product.

There are two types of record used to identify Global Auxiliary Data. A Global External Auxiliary Data Record identifies GAD external to the product by means of a pointer that uniquely identifies the data. A Global Internal Auxiliary Data Record contains GAD data that has been embedded within the product itself.

The GAD section may contain zero or more GAD records.

The GAD records are ordered first by record class and then by increasing record subclass.

Global Auxiliary Data Section
Global External Auxiliary Data Records
Global Internal Auxiliary Data Records

Table 33: Breakdown of Auxiliary Data Section

5.3.1 Global External Auxiliary Data Record (GEADR)

Each Global External Auxiliary Data Record (GEADR) will contain a pointer consisting of an ASCII string of up to 100 characters that uniquely identifies an external auxiliary dataset. This uniqueness includes being able to uniquely identify updates of the same dataset. If the dataset name follows a convention that allows such unique identification, then this dataset name should be used. If the pointer is less than 100 characters in length, ASCII space characters should be appended to the end of the pointer to pad the length to 100 characters.

There is one GEADR per each external auxiliary dataset. The type of auxiliary dataset referenced is determined by the Record Subclass. Any individual GEADR will appear only once per product.

A GEADR must be generated whenever a global external auxiliary dataset is used by the processor, even if it is not explicitly stated in a given instrument PFS document that the product contains GEADRs.

The format of the GEADR is described in the Annex (Appendix A) to this document.

5.3.1.1 Generic Record Header Fields

The GEADR has the following GRH fields:

Field	Value
RECORD_CLASS	GEADR
INSTRUMENT_GROUP	<see relevant PFS>
RECORD_SUBCLASS	<see relevant PFS>

Table 34: Generic Record Header fields for GEADR

The record subclass indicates the type of auxiliary data pointed to by the record.

5.3.2 Global Internal Auxiliary Data Record (GIADR)

The Global Internal Auxiliary Data Record (GIADR) contains information required for the complete product that can be embedded in the actual product. On the whole, this information will be instrument and/or level specific and so there will be a number of record subclassses of GIADRs each defined in the relevant PFS document.

Any individual GIADR will appear only once per product.

5.3.2.1 Generic Record Header Fields

The GIADR has the following GRH fields:

Field	Value
RECORD_CLASS	GIADR
INSTRUMENT_GROUP	<see relevant PFS>
RECORD_SUBCLASS	<see relevant PFS>

Table 35: Generic Record Header fields for GIADR

The record subclass indicates the type of auxiliary data contained in the record.

5.4 Variable Auxiliary Data Section

The Variable Auxiliary Data (VAD) is any auxiliary data associated with the product processing that is or may be updated during the processing of the product. This data may be generated by the processing function itself or may be acquired through the PGE. Unlike the GAD, the VAD record is only applicable until it is superseded by another VAD record of the same subclass.

The data in a VAD is updated less frequently than the MDR occurrence rate.

There are two types of record used to identify Variable Auxiliary Data. A Variable External Auxiliary Data Record identifies VAD external to the product by means of a pointer that uniquely identifies the data. A Variable Internal Auxiliary Data Record contains VAD data that has been embedded within the product itself.

The VAD section may contain zero or more VAD records. The VAD records are ordered first by record class and then by increasing record subclass.

Variable Auxiliary Data Section
Variable External Auxiliary Data Records
Variable Internal Auxiliary Data Records

Table 36: Breakdown of Variable Auxiliary Data Section

5.4.1 Variable External Auxiliary Data Record (VEADR)

A Variable External Auxiliary Data Record (VEADR) is very similar to a GEADR, except that the information it points to is not necessarily relevant to the entire product but may be updated during the time period covered by the product, thus there may be one or more of each VEADR subclass within a product, valid for different times (as identified in the GRH).

Each VEADR will contain a pointer consisting of an ASCII string of up to 100 characters that uniquely identifies an external auxiliary dataset. This uniqueness includes being able to uniquely identify updates of the same dataset. If the dataset name follows a convention that allows such unique identification, then this dataset name should be used. If the pointer is less than 100 characters in length, ASCII space characters should be appended to the end of the pointer to pad the length to 100 characters.

The type of auxiliary dataset referenced is determined by the Record Subclass.

A VEADR must be generated whenever a global external auxiliary dataset is used by the processor, even if it is not explicitly stated in a given instrument PFS document that the product contains VEADRs.

The format of the VEADR is described in the Annex (Appendix A) to this document.

5.4.1.1 Generic Record Header Fields

The VEADR has the following GRH fields:

Field	Value
RECORD_CLASS	VEADR
INSTRUMENT_GROUP	<see relevant PFS>
RECORD_SUBCLASS	<see relevant PFS>

Table 37: Generic Record Header fields for VEADR

The record subclass indicates the type of auxiliary data pointed to by the record.

5.4.2 Variable Internal Auxiliary Data Record (VIADR)

A VIADR is very similar to a GIADR, except that the information it points to is not necessarily relevant to the entire product but may be updated during the time period covered by the product, thus there may be one or more of each VIADR subclass within a product, valid for different times (as identified in the GRH).

On the whole, this information will be instrument and/or level specific and so there will be a number of record subclasses of VIADRs each defined in the relevant PFS document.

5.4.2.1 Generic Record Header Fields

The VIADR has the following GRH fields:

Field	Value
RECORD_CLASS	VIADR
INSTRUMENT_GROUP	<see relevant PFS>
RECORD_SUBCLASS	<see relevant PFS>

Table 38: Generic Record Header fields for VIADR

The record subclass indicates the type of auxiliary data contained in the record.

5.5 Body Section

Body Section
Measurement Data Records

Table 39: Breakdown of Body Section

The Body Section contains time-ordered Measurement Data Records (MDRs)

5.5.1 Measurement Data Record (MDR)

The Measurement Data Record is level and instrument dependent. For Level 1 data, it contains the instrument measurements including the earth view as well as calibration target measurements. There is usually one MDR per scan line (or per across track line for ASCAT, or per occultation for GRAS). For Level 2 data products, the MDR contains the measurements derived from the instrument measurements e.g. temperature or water vapour profiles.

Details of the MDR are given in the relevant PFS documents.

5.5.1.1 Generic Record Header Fields

The MDR has the following GRH fields:

Field	Value
RECORD_CLASS	MDR
INSTRUMENT_GROUP	<see relevant PFS>
RECORD_SUBCLASS	<see relevant PFS>

Table 40: Generic Record Header fields for MDR

The record subclass indicates the type of measurement data contained in the record.

5.5.2 Dummy Measurement Data Record (DMDR)

The Dummy Measurement Data Record is a special case of the MDR. It is a generic record that is used to indicate the location of lost data within any product. It contains a STATUS_FLAG which can be used to indicate the reason for the lost data. One DMDR can replace a single lost MDR or a contiguous block of lost MDRs. Thus it is possible to have a number of contiguous DMDRs that may have the same or differing STATUS_FLAG settings and which may each replace 1 or more lost MDRs.

The DMDR can be recognised from other MDRs by its INSTRUMENT_GROUP value of DUMMY which is unique to the DMDR.

The DMDR is a subclass of MDR and therefore should be treated as an MDR.

A DMDR is generated whenever there is data on the scale of an MDR or greater absent from a product and where the data loss could not be easily predicted by the end-user e.g. loss due to data dropout, timeliness violation, moving to an instrument mode that is not part of the normal observation cycle, etc. In contrast, data that is routinely missing should not generate a DMDR e.g. the normal loss of two lines of HIRS data per every 40 scan lines due to calibration activities.
This rule may be over-ridden for specific cases for an instrument, as detailed in the PGS and/or PFS for that instrument.

5.5.2.1 Generic Record Header Fields

The DMDR has the following GRH fields:

Field	Value
RECORD_CLASS	MDR
INSTRUMENT_GROUP	DUMMY
RECORD_SUBCLASS	1

Table 41: Generic Record Header fields for DMDR

The RECORD_START_TIME and RECORD_STOP_TIME for the DMDR correspond to the times of the missing MDRs it replaces, with DMDR.RECORD_START_TIME corresponding to the RECORD_START_TIME of the first missing MDR, and DMDR.RECORD_STOP_TIME set to the RECORD_STOP_TIME of the last missing MDR.

For instruments that produce data at regular intervals, these times can be calculated from the repeat cycle.

For instruments that produce data at irregular rates, the mechanism for calculating these times is to set the DMDR start time to the preceding MDR or DMDR end time plus one millisecond, and to set the DMDR end time to the following MDR or DMDR start time minus one millisecond, unless instructed to do otherwise by the relevant PGS or PFS. Where a DMDR is not preceded by an MDR, then the start theoretical time in the MPHR is used to set the DMDR start time. Where a DMDR is not followed by an MDR then the end theoretical time in the MPHR is used to set the DMDR end time.

5.5.2.2 DMDR Fields

5.5.2.2.1 STATUS_FLAG

This field contains a status flag of type ENUMERATED.

Value	Meaning
0	Default Value – No reason given for loss of data
1 – 255	TBD

Table 42: STATUS_FLAG enumerated values

5.6 Record Format Version Control

This section provides current and previous version numbers for the generic records defined within this document.

Record Subclass	Format Version Number	Issue Defined
MPHR	2	6.3
	1	6.2 (CDR)
GEADR	1	6.2 (CDR)
VEADR	1	6.2 (CDR)
DUMMY-MDR	2	6.3
	1	6.2 (CDR)
IPR	1	6.2 (CDR)

Table 43: Record Format Version Numbers for Generic Records

6 FULL PRODUCTS, REGIONAL PRODUCTS AND PDUS

6.1 Terminology

EPS products are listed in the CGSRD. They are produced by a Product Generation Function (PGF). All EPS products are constructed from components that are described in this document.
The generic formats of the EPS products are described by this document. Instrument and Processing Level specific issues are covered in the relevant PFS document.

The Full EPS product is produced by processing a dump of data. Note that, dependent upon the instrument and processing level, the derived full product may cover a period of time that is less or more than the period of time covered by the original dump from which it is derived. This is described in the relevant PGS documents.

In addition to the Full product, there are also two sub-products, the Regional EPS product and the Product Dissemination Unit (PDU), both of which follow the same format specification as the parent full product.

The Regional Product is a full product that has been passed through a geographic filter. It may be produced either by an NRT terminal, in which case the level of granularity for the filtering is the PDU) or by the archive facility UMARF, in which case the level of granularity for the filtering is the MDR.

The Product Dissemination Unit is the dissemination unit for the full product. In order to maintain the timeliness requirements, [**AD-49**] requires that an EPS product is disseminated via a Near Real Time (NRT) terminal in small units called Product Dissemination Units (PDUs). The PDUs are reassembled into the full product, or one or more regional products, by the NRT terminal [**AD-49**].

For a PDU, the Orbit Start and Orbit End fields in the MPHR will differ from the standard GPFS Full product specification. They will contain the start and end orbit number of the dump from which the PDU was derived, and not the start and end orbit numbers of the PDU itself.

For a PDU, the Theoretical Sensing Start and End fields in the MPHR will differ from the standard GPFS Full product specification. They will contain the theoretical start and end sensing times of the dump from which the PDU was derived, and not the times for the PDU itself. This allows missing data at the start and end of a reconstructed product to be detected.

SECTION	RECORD CLASS	RECORD SUBCLASS	START TIME	STOP TIME
HEADER SECTION	MAIN PRODUCT HEADER RECORD		T_1	T_6
	SECONDARY PRODUCT HEADER RECORD		T_1	T_6
INTERNAL POINTER SECTION	INTERNAL POINTER RECORD (GEADR Subclass A)		T_1	T_6
	INTERNAL POINTER RECORD (GEADR Subclass B)		T_1	T_6
	INTERNAL POINTER RECORD (GIADR Subclass A)		T_1	T_6
	INTERNAL POINTER RECORD (GIADR Subclass B)		T_1	T_6
	INTERNAL POINTER RECORD (GIADR Subclass C)		T_1	T_6
	INTERNAL POINTER RECORD (VEADR Subclass A)		T_1	T_6
	INTERNAL POINTER RECORD (VEADR Subclass B)		T_1	T_6
	INTERNAL POINTER RECORD (VEADR Subclass C)		T_1	T_6
	INTERNAL POINTER RECORD (VIADR Subclass A)		T_1	T_6
	INTERNAL POINTER RECORD (VIADR Subclass B)		T_1	T_6
	INTERNAL POINTER RECORD (VIADR Subclass C)		T_1	T_6
	INTERNAL POINTER RECORD (MDR Subclass A)		T_1	T_6
	INTERNAL POINTER RECORD (MDR Subclass B)		T_1	T_6
	INTERNAL POINTER RECORD (MDR DUMMY)		T_1	T_6
	INTERNAL POINTER RECORD (MDR Subclass A)		T_1	T_6
	INTERNAL POINTER RECORD (MDR Subclass B)		T_1	T_6
GLOBAL AUXILIARY DATA SECTION	GLOBAL EXTERNAL AUXILIARY DATA RECORD	SUBCLASS A	T_1	T_6
	GLOBAL EXTERNAL AUXILIARY DATA RECORD	SUBCLASS B	T_1	T_6
	GLOBAL INTERNAL AUXILIARY DATA RECORD	SUBCLASS A	T_1	T_6
	GLOBAL INTERNAL AUXILIARY DATA RECORD	SUBCLASS B	T_1	T_6
	GLOBAL INTERNAL AUXILIARY DATA RECORD	SUBCLASS C	T_1	T_6
VARIABLE AUXILIARY DATA SECTION	VARIABLE EXTERNAL AUXILIARY DATA RECORD	SUBCLASS A	T_1	T_6
	VARIABLE EXTERNAL AUXILIARY DATA RECORD	SUBCLASS B	T_1	T_3
	VARIABLE EXTERNAL AUXILIARY DATA RECORD	SUBCLASS B	T_3	T_6
	VARIABLE EXTERNAL AUXILIARY DATA RECORD	SUBCLASS C	T_1	T_5
	VARIABLE EXTERNAL AUXILIARY DATA RECORD	SUBCLASS C	T_5	T_6
	VARIABLE INTERNAL AUXILIARY DATA RECORD	SUBCLASS A	T_1	T_2
	VARIABLE INTERNAL AUXILIARY DATA RECORD	SUBCLASS A	T_2	T_4
	VARIABLE INTERNAL AUXILIARY DATA RECORD	SUBCLASS A	T_4	T_6
	VARIABLE INTERNAL AUXILIARY DATA RECORD	SUBCLASS B	T_1	T_6
	VARIABLE INTERNAL AUXILIARY DATA RECORD	SUBCLASS C	T_1	T_6
BODY SECTION	MEASUREMENT DATA RECORD	SUBCLASS A	T_1	T_2
	MEASUREMENT DATA RECORD	SUBCLASS B	T_2	T_3
	MEASUREMENT DATA RECORD	DUMMY	T_3	T_4
	MEASUREMENT DATA RECORD	SUBCLASS A	T_4	T_5
	MEASUREMENT DATA RECORD	SUBCLASS B	T_5	T_6

Figure 2: Generalised Schematic of the Generic Product Format.

The number, timing and frequency of multiply occurring records is for illustrative purposes only. Shading in the Record Class column is to highlight record class groupings. Shading in the Record Subclass column is to highlight record subclass groupings. Shading in the Start/Stop Time columns is to highlight timings across record subclass groupings. For descriptions of how stop/start times for various records are derived and their meanings, see Section 4.2.

7 PRODUCT NAMING CONVENTION

EPS products shall be named using the following naming convention, which provides a product name that uniquely identifies any product and provides a summary of its contents.

The name will be composed of a number of product name fields separated by underscore characters, "_", as shown below.

> **<INSTRUMENT_ID>_<PRODUCT_TYPE>_<PROCESSING_LEVEL>_<SPACECRAFT_ID>_**
> **<SENSING_START>_<SENSING_END>__<PROCESSING_MODE>_<DISPOSITION_MODE>**
> **_<PROCESSING_ TIME>**

Each product name field is directly related to the field of the same name within Main Product Header Record (MPHR) as shown in Table 44. The fields may thus contain only upper case letters [A…Z], lower case 'x' (to represent a blank item), and numbers [0…9].

As the size of each product name field is fixed, the product name is always 67 characters in length.

Example product name:

```
AMSU_xxx_1A_M01_20050101101500Z_20050101115500Z_N_O_20050101121500Z
```

Product Name Field / MPHR Field	**Description**	**Size in Characters**
INSTRUMENT_ID	Instrument identification	4
PRODUCT_TYPE	Product Type	3
PROCESSING_LEVEL	Processing Level Identification	2
SPACECRAFT_IUD	Spacecraft identification	3
SENSING_START	UTC Time of start of Sensing Data	15
SENSING_END	UTC Time of end of Sensing Data	15
PROCESSING_MODE	Identification of the mode of processing	1
DISPOSITION_MODE	Identification of the type of processing	1
PROCESSING_TIME	UTC time at start of processing for the product	15

Table 44: Derivation of Product Name Fields from the Main Product Header Record

8 PRODUCT FORMAT VERSION CONTROL

As presented in Section 4.2.1.1.4, each record subclass has an associated record subclass format version number in the RECORD_SUBCLASS_VERSION field of the generic record header. In addition, each product has a format version number (consisting of a major and minor version number, each with allowed values running from 0 to 99999) which is stored in the MPHR fields FORMAT_MAJOR_VERSION and FORMAT_MINOR_VERSION.

Record format version numbers are recorded in the relevant PFS document. Product format version numbers are recorded in [**RD-223**].

The product format version number should be updated whenever there is a change in the format or contents of a product that requires an update to software that has to read the product or has to check the product is assembled correctly from the component records. This could be a change in the format itself (record field deleted, added, resized, retyped), a change in the contents of a field (e.g. scale factor change) or a change in the way that a field has to be interpreted. Any such record update requires the record format version number to be incremented. So, the updating of a record necessarily implies an updating of the format of any product that utilises the record, necessitating an update of the product format version number.

In addition, if a product no longer contains a certain record that was once compulsory, or adds a new record, or changes the way in which a record is used, then there should also be a new product format version number.

To summarise, the product format version number is updated:

1. when any record format version number of a record used in that product changes,
2. when a compulsory record is removed from a product,
3. when a new record is added to the product,
4. when the use of a record changes (an example here might be dropping dummy MDRs).

The *major.minor* versions of the product format version number were previously used to indicate products that fell between established baselines. For example, 3.0 and 4.0 would correspond to particular baselines (e.g. an issued set of GPFS/PFS documents) whereas 3.1 might correspond to a new GPFS and previous PFS, or to the partial implementation of the latest PFS. A recommended way to use both version numbers in the future is to issue minor updates for a change resulting from a PFS update, and major updates for a change resulting from GPFS updates that affect all products. Then a GPFS update would reset all products back to a new major of (say) 12.0, and then 12.1, 12.2 etc. versions would indicate PFS-only updates.

9 EPS PRODUCT GENERIC LEVEL 0 FORMAT

9.1 Overview

The Level 0 product shall contain the raw instrument data. This data shall be present in the product in a format that retains all required information from the original raw data in a format that resembles the original raw data as closely as possible, within the overall design philosophy constraints of the system.

For Metop data, the Level 0 products for each instrument shall have instrument source packets contained in MDRs.

For NOAA data, the Level 0 products for all instruments shall have GAC frames contained in MDRs.

The following Sections describe those components of the Level 0 product that differ from or are not covered by the generic product format specification.

9.2 Secondary Product Header Record

There is no SPHR defined for the L0 products.

9.3 Variable Internal Auxiliary Data Records

There is one VIADR that is common to all Level 0 Metop products. It contains the OBT-UTC correlation parameters for the product. The format of this VIADR can be found in the Annex (Appendix A) to this document.

This Level 0 OBT2UTC VIADR normally occurs 1 or more times per product. However, in certain non-nominal situations (e.g. a missing time correlation auxiliary data file or an unusable administrative message) the VIADR may be dropped from the Level 0 product. In this case, the onboard UTC time in the ISPs may be used as the best available time tag.

9.3.1 Generic Record Header Fields

The Level 0 VIADR has the following GRH fields:

Field	Value
RECORD_CLASS	VIADR
INSTRUMENT_GROUP	GENERIC
RECORD_SUBCLASS	0

Table 45: Generic Record Header fields for Level 0 VIADR

9.4 Measurement Data Records

For a Metop Level 0 product, the INST_DATA field of the MDR contains either an instrument source packet (ISP) or, in the case of the HKTM product, a satellite source packet (SSP). These source packets are defined in **[RD-1]**.

For a NOAA Level 0 product, the INST_DATA field contains a NOAA GAC, STIP or SAIP packet as defined in **[RD-107].** Note that this data contains the integrated NOAA instruments, so there is only one L0 NOAA product (as opposed to Metop, which has a L0 product per instrument).

Although the format of the Level 0 MDR is generic, the size of the INST_DATA field will vary depending upon platform (Metop/NOAA), instrument and measurement data type. The size of the INST_DATA byte array is contained in the INST_DATA_SIZE field.

The contents of any given MDR can be identified by a combination of the RECORD_INSTRUMENT_GROUP and RECORD_SUBCLASS fields within the generic record header.

The raw instrument data contained within the INST_DATA field will be a direct copy of the binary information from the ISP or GAC packet. This data is therefore not formatted according to GPFS specifications (for example, it may contain values of a real type) and must be interpreted according to the relevant instrument telemetry specifications [**RD-1**, **RD-107** and additional references therein].

The format of the Level 0 MDR can be found in the Annex (Appendix A) to this document.

9.4.1 Generic Record Header Fields

The Level 0 MDR has the following GRH fields:

Field	Value
RECORD_CLASS	MDR
INSTRUMENT_GROUP	GENERIC
RECORD_SUBCLASS	See Table 47

Table 46: Generic Record Header fields for Level 0 MDR

Value	Meaning
0	Metop data – MDR contains an ISP
1	NOAA data – MDR contains a GAC frame
2	NOAA data – MDR contains an AIP frame
3	NOAA data – MDR contains a TIP frame
4	Metop data – MDR contains an SSP

Table 47: Record Subclass values for GRH in Level 0 MDR

As described in Table 15, the GRH field RECORD_START_TIME is set to the sensing start of the first measurement in the record. The RECORD_STOP_TIME is normally set to the sensing stop of the last measurement in the record, but as this information is not readily available for Level 0 MDRs the RECORD_STOP_TIME is set equal to the RECORD_START_TIME.

9.4.2 Degraded Instrument MDR Flag

The DEGRADED_INST_MDR flag shall be set whenever a Level 0 MDR is of reduced quality due to a manoeuvre.

9.5 Record Format Version Control

This section provides version numbers for the Level 0 records defined within this document.

Record Subclass	Format Version Number	Issue Defined
SPHR-L0	DELETED	6.3
	1	6.2 (CDR)
VIADR-L0-OBT2UTC	2	6.3
	1	6.2 (CDR)
MDR-L0	1	6.2 (CDR)

Table 48: Record Format Version Numbers for Generic Level 0 Records

APPENDIX A RECORD TEMPLATES

The following Annex provides detailed record format descriptions of generic records and records specific to Level 0 products, i.e.:

- MPHR
- IPR
- GEADR
- VEADR
- DMDR
- VIADR-L0-OBT2UTC
- MDR-L0

The Annex is accessible under Document Reference: EPS.GGS.SPE.96167.ANX
or electronically via the following Hummingbird link:

DOCSLIB-#213604-Generic Product Format Specification - Annex

This Document	
Title	EPS Generic Product Format Specification - ANNEX
Reference Number	EPS.GGS.SPE.96167

Revisions	
Issue 4 Draft B	Updated Excel spreadsheet layout
	Added RECORD_NUMBER to REC_HEAD compound type
	Changed RECORD_INSTANCE to RECORD_SUBCLASS to avoid confusion with OO naming conventions
	Moved Level 0 records from main document to this annex
	Added ACTUAL_PRODUCT_SIZE field to MPHR
	Rearranged fileds in MPHR to bring like items together
	Renamed PROC_TIME to PROC_TIME_START and added PROC_TIME_END to MPHR
Draft C	Added PRODUCT_NAME field to MPHR
	Added PARENT_PRODUCT_NAME field to MPHR
	Changed INST_MODEL in MPHR to enumerated field type
	Added ASCII version of MPHR
	Removed MPHR fields: PRODUCT_CONF and PROC_QUAL
	Removed TIME_SEQUENCE_CODE, EARTH_LOCATION_ERROR and EARTH_LOCATION_ERROR from MPHR
	Added following fields to MPHR: MEAS_START_LAT/LON, MEAS_STOP_LAT/LON
	Removed Level 0 VIADRs
	Combined MetOp and NOAA level 0 MDR into one generic MDR
Draft D	Removed SRC_DATA_QUAL field from MPHR (non-generic). Place in SPHRs if required.
	Removed binary version of MPHR
Issue 5 Revision 0	Issue for CGS PDR
Issue 5 Revision 1	Revised Issue for CGS PDR
Issue 5 Revision 2	This Annex was missing from the hardcopy distribution of Issue 5 Rev 1
Issue 5 Revision 3	Added total count of MPHR, SPHR and IPR to the MPHR
	Expanded names of fields in MPHR to make them more human-readable and consistent as these will appear as ASCII in all products
	MPHR field "ACTUAL_PRODUCT_SIZE" changed to u-integer4 from u-integer8
	IPR updated to contain table to all records, not just first of a class
	Fields containing Product Names now set at 83 characters
Issue 6 Revision 0	Update IPR to contain a table of pointers for all records of a given class. One IPR per product per record class
	Add compound type POINTER

	Added DMDR (simply a correctly flagged GRH)
	Removed record numbers from GHR - times used to reference records.
	Updated fields that refer to record numbers so that times are used instead
	Delete TOTAL_COUNT_ADRS from MPHR - no longer have ADRs
	Updated counts of record types in MPHR to use new flags that each MDR should contain.
	Added PDU information to MPHR
	Added Regional filtering information to MPHR
	Added Archive information to MPHR
	Modified and added Error flag information in the Level 0 MDR
	Add MPHR.SENSING_START_DUMP, SENSING_END_DUMP fields to cater for products that contain infomration beyond the duration of the main dump on which the full product is based
	Add MPHR.SENSING_START_FULL_PRODUCT, SENSING_END_FULL_PRODUCT fields
	Moved OBT to UTC conversion data into MPHR from L0 SPHR
	Moved leap second data into MPHR from L0 SPHR
	MPHR.PROCESSING_LEVEL size corrected from e-char(3) to e-char(2)
	MPHR Field names made more readable
	MPHR increaseed parent product names to 4 to account for ATOVS L2 and other products where more than one product is a parent
Issue 6 Revision 1	Removed fields MPHR.SENSING_START_FULL_PRODUCT, SENSING_END_FULL_PRODUCT
	Removed fields MPHR.MEAS_LAT/LON_START/STOP
	Updated definitin of POINTER compound. Removed TARGET_RECORD_SIZE/STARTS_TIME/STOP_TIME/STATUS
	Removed array of pointers from IPR and replaced with a single pointer to the first record of the relevant record class
	Deleted MPHR.MDR_RATE and MPHR.MDR_DURATION fields
	Added MPHR.CLOCK_UPDATE and CLOCK_UPDATE_TIME
	Removed MPHR.COUNT_MDR_MISSING_AUX_DATA
	Removed MPHR.COUNT_MISSING_MDR and COUNT_MISSING_MDR_GAPS
	Major changes to Level 0 SPHR. Removed redundant fields. Renamed fields to better describe actual contents.
	Removed fields from SPHR L0: TOTAL_VCDUS, TOTAL_DISCARDED_ISPS
	Replaced L0 SPHR.TOTAL_PARITY_CHECK_ERRORS with TOTAL_PEC_ERRORS
	Deleted L0 MDR fields: COUNT_VCDUS_FOR_ISP, COUNT_RS_DECODE_FAULTS, COUNT_APID_SEQUENCE_ERRORS,COUNT_CCSDS_SEQUENCE_ERRORS
	Renamed L0 MDr.PARITY_CHECK_ERROR_COUNT to PEC_ERROR_CHECK to cover both Vertical Parity Check and Cyclic Redundancy Check (mutually exclusive)

	Added fields to MPHR: DURATION_OF_PRODUCT, MILLISECONDS_OF_DATA_PRESENT, MILLISECONDS_OF_DATA_MISSING
	Updated ascending node orbit parameters in MPHR. Changed types where necessary to allow required accuracy. Added scaling factors and units where appropriate
	Split MPHR.LOCATION_TOLERANCE into three fields as it has three components but arrays are not allowed in ASCII fields
	Deleted MPHR.LOCATION_FLAG
	Deleted MPHR.SUBSETTED_PRODUCT_FLAG and SUBSET_TIME
	Updated product names fields in MPHR to 67 characters
	Deleted SPHR L0 Fields: COUNT_ACQUISITION_STATION_PARITY_ERROR, SUM_AUXILIARY_SYNCH_ERROR, TIME_SEQUENCE_ERROR, TIME_SEQUENCE_ERROR_TIME, TIME_SEQUENCE_CODE, ACQUISITION_STATION_CLOCK, ACQUISITION_STATION_CLOCK_TIME, EARTH_LOCATION_INDICATOR, EARTH_LOCATION_ERROR
	Removed RECORD_STATUS from GRH. Added MISSING_DATA_FLAG to DMDR to fufill same purpose of indicating reason for missing data.
Issue 6 Revision 2 DRAFT	Updated Leap Second fields in MPHR. LEAP_PRESENCE and LEAP_SIGN fields combined into single LEAP_SECOND field (which will take values of -1, 0 or +1 depending upon presence of leap second occurence within this product and sign). Change from 6 characters to 2 characters in size.
	LEAP_UTC renamed to LEAP_SECOND_UTC for clarity. Change field type from short cds time to general time as short cds time can't be used in an ASCII record. Increase type size from 6 characters to 15 characters.
	Added GIADR-L0-OBT2UTC to contain information necessary to perform OBT to UTC conversions
	Added OB clock update information to MPHR
	MPHR fields SENSING_START_DUMP and SENSING_END_SUMP renamed to SENSING_START_THEORETICAL and SENSING_END_THEORETICAL in order to clarify their contents and use.
	In ASCII records, change TYPE column to EQUIVALENT TYPE as a better description
	Update defintion of EARTH_SUN_DISTANCE_RATIO. Correct units from km to NA as it is a ratio
	Remove TBCs from MPHR fields - ROI_FILTERED and ARCHIVE_RETRIEVAL
	MPHR field SOURCE_GROUND_STATION renamed to RECEIVING_GROUND_STATION to clarify the function of the field
	Removed followiong redundant fields from Level 0 MDR - PEC_ERROR_CHECK, FRAME_SYNC_ERROR, PARITY_ERROR_CHECK, AIP_SYNC_ERROR, TIP_SYNC_ERROR, AIP_SEQUENCE_ERROR, TIP_SEQUENCE_ERROR, INST_DATA_CORRUPT
	DMDR: Rename MISSING_DATA_FLAG to LOST_DATA_FLAG for consistency
	All fields in MPHR subsections - ASCENDING NODE ORBIT PARAMETERS and LOCATION SUMMARY made 11 characters in size

Issue 6 Revision 2	GIADR-L0-OBT2UTC changed to VIADR-L0-OBT2UTC as OBT-UTC update normally occurs during a product so there may be different parameters applied to different parts of the product
	Updated VEADR and GEADR AUX_DATA_POINTER field to make them 100 characters in length
	Changed MPHR.STATE_VECTOR_TIME data type to LONG GENRAL TIME to give millisecond accuracy
Issue 6 Revision 3	Updated Annex based on EUM.EPS.SYS.DCR.02.143
	Correct POINTER size in Types sheet from 19 to 7. Affects IPR sheet.
	Increase size of fields MPHR.DURATION_OF_PRODUCT, MILLISECONDS_OF_DATA_PRESENT and MILLISECONDS_OF_DATA_MISSING from 6 to 8 characters
	Added text to descriptiuon of MPHR.SENSING_START_THEORETICAL and SENSING_END_THEORETICAL
	Removed fields MPHR.OBT_UTC_CORRECTION and OBT_UTC_CORRECTION_UTC
	Add text to desription of MDR-L0.DEGRADED_INST_MDR and DEGRADED_PROC_MDR
	Updated description for MDR-L0.INST_DATA
	Deleted SPHR-L0
	Corrected Offset Calculations where necessary
	DMDR.LOST_DATA_FLAG renamed to SPARE_FLAG and description updated
	MPHR State Vector Components type changed from U-INTEGER to INTEGER
	Update text description for VIADR-L0-OBT2UTC fields
	Delete field MPHR.ARCHIVE_RETRIEVAL
	Rename MPHR.ROI_FILTERED to SUBSETTED_PRODUCT. Update description.
Issue 6 Revision 4	**EUM.EPS.SYS.DCR.03.011**
	Changes to text of main document
Issue 6 Revision 5	**EUM.EPS.SYS.DCR.04.0082**
	Update to VIADR-L0-OBT2UTC
	Field UTC_OBT_TIME renamed to UTC_0 and changed definition of contents
	Field OBT_UTC_TIME renamed to CCU_OBT_0 and changed defintion of contents
	Field CLOCK_STEP changed definition of contents
	EUM.EPS.SYS.DCR.05.0192
	Added indication of relevant reference frame (as defined in the Mission Conventions Document [AD-48]) to orbit state vector field descriptions in the MPHR
Issue 6 Revision 6	**EUM.EPS.SYS.DCR.05.0254**
	No change to Annex
Version 7A 22/08/08	Migrated into Hummingbird. Contents identical with Issue 6.6.

Version 7B 29/01/09	Inclusion of updates made for Issue 6.7 Draft, originally made to document before it was put into Hummingbird. Details:
	EUM.SYS.DCR.07.0343
	No change to Annex
	EUM.SYS.DCR.07.0348
	No change to Annex
	EUM.SYS.DCR.05.0255
	MPHR > EARTH_SUN_DISTANCE_RATIO. Added scale factor 6. Added referenced value for 1 AU.
	EUM.SYS.DCR.05.0290
	IPR > POINTER - change to description. Remove text : ", as specified in the subclass of the IPR generic record header"
	EUM.SYS.DCR.06.0341
	DUMMY MDR > Rename SPARE_FLAG to STATUS_FLAG and change description from "Not currently used"
	EUM.SYS.DCR.07.0358
	MPHR> ORBIT_START and ORBIT_END. Extend description to clarify definition of time at which start and end orbits are determined
	EPS.AB.DCR.EUM.1
	No change to Annex
	EPS.AB.DCR.EUM.11
	Use of DEGRADED_INST_MDR flag in L0 MDRs.
	EPS_AB_ACTION_6417
	No change to Annex
Version 7C 20/10/09	**ODT_DCR_100, ODT_DCR_101**
	No change to Annex
Version 7D 17/11/09	**ODT_DCR_104**
	No change to Annex
Version 7E 30/11/10	**ODT_DCR_134**
	No change to Annex
	ODT_DCR_220
	MPHR: Update to descriptions of SENSING_START and SENSING_END fields.

	REC_HEAD - Generic Record Header									
FIELD	**DESCRIPTION**	**SF**	**UNITS**	**DIM1**	**DIM2**	**DIM3**	**TYPE**	**TYPE SIZE**	**FIELD SIZE**	**OFFSET**
RECORD_CLASS	Class of Record	0	N/A	1	1	1	enumerated	1	1	0
INSTRUMENT_GROUF	Defining group for record subclasses	0	N/A	1	1	1	enumerated	1	1	1
RECORD_SUBCLASS	Subclass of this record class	0	N/A	1	1	1	enumerated	1	1	2
RECORD_SUBCLASS_VERSION	Version of this particular format of the record case	0	N/A	1	1	1	enumerated	1	1	3
RECORD_SIZE	Total size of the record case (including this header)	0	N/A	1	1	1	u-integer4	4	4	4
RECORD_START_TIME	Start Time for this record - context will depend on record class	0	N/A	1	1	1	short cds time	6	6	8
RECORD_STOP_TIME	Stop Time for this record - context will depend on record class	0	N/A	1	1	1	short cds time	6	6	14
									TOTAL	20

	POINTER - Generic Record Pointer									
FIELD	**DESCRIPTION**	**SF**	**UNITS**	**DIM1**	**DIM2**	**DIM3**	**TYPE**	**TYPE SIZE**	**FIELD SIZE**	**OFFSET**
TARGET_RECORD_CLASS	Class of target record as derived from the GRH of the target record	0	N/A	1	1	1	enumerated	1	1	0
TARGET_INSTRUMENT_GROUP	Defining group for target record subclass as derived from the GRH of the target record	0	N/A	1	1	1	enumerated	1	1	1
TARGET_RECORD_SUBCLASS	Subclass of target record class as derived from the GRH of the target record	0	N/A	1	1	1	enumerated	1	1	2
TARGET_RECORD_OFFSET	Offset of target record from start of product	0	Bytes	1	1	1	u-integer4	4	4	3
									TOTAL	7

FIELD	DESCRIPTION	SF	UNITS	EQUIVALENT TYPE	ENCODE CHARS	FIELD SIZE	OFFSET
RECORD_HEADER	Generic Record Header - NOTE: This is binary!	NA	NA	REC_HEAD	20	20	0
Product Details							
PRODUCT_NAME	Complete name of the product	NA	NA	*CHAR*	67	100	20
PARENT_PRODUCT_NAME_1	Name of the parent product from which this product has been produced. For Level 0 products, this field is filled with lower case x's.	NA	NA	*CHAR*	67	100	120
PARENT_PRODUCT_NAME_2	Name of the parent product from which this product has been produced. For Level 0 products or products for which this is not appropriate, this field is filled with lower case x's.	NA	NA	*CHAR*	67	100	220
PARENT_PRODUCT_NAME_3	Name of the parent product from which this product has been produced. For Level 0 products or products for which this is not appropriate, this field is filled with lower case x's.	NA	NA	*CHAR*	67	100	320
PARENT_PRODUCT_NAME_4	Name of the parent product from which this product has been produced. For Level 0 products or products for which this is not appropriate, this field is filled with lower case x's.	NA	NA	*CHAR*	67	100	420
INSTRUMENT_ID	Instrument identification	NA	NA	*E-CHAR*	4	37	520
INSTRUMENT_MODEL	Instrument Model identification	NA	NA	*ENUMERATED*	3	36	557
PRODUCT_TYPE	Product Type	NA	NA	*E-CHAR*	3	36	593
PROCESSING_LEVEL	Processing Level Identification	NA	NA	*E-CHAR*	2	35	629
SPACECRAFT_ID	Spacecraft identification	NA	NA	*E-CHAR*	3	36	664
SENSING_START	UTC start time of sensing data in this object (PDU, ROI or Full Product) of original product as planned by the PROCESSING_CENTRE	NA	NA	*GENERAL TIME*	15	48	700
SENSING_END	UTC end time of sensing data in this object (PDU, ROI or Full Product) of original product as planned by the PROCESSING_CENTRE	NA	NA	*GENERAL TIME*	15	48	748
SENSING_START_THEORETICAL	Theoretical UTC Time of start of sensing data in the dump from which this object is derived. This data is the predicted start time at the MPF level.	NA	NA	*GENERAL TIME*	15	48	796
SENSING_END_THEORETICAL	Theoretical UTC Time of end of sensing data in the dump from which this object is derived. This data is the predicted end time at the MPF level.	NA	NA	*GENERAL TIME*	15	48	844
PROCESSING_CENTRE	Processing Centre Identification	NA	NA	*E-CHAR*	4	37	892
PROCESSOR_MAJOR_VERSION	Processing chain major version number	NA	NA	*U-INTEGER*	5	38	929
PROCESSOR_MINOR_VERSION	Processing chain minor version number	NA	NA	*U-INTEGER*	5	38	967
FORMAT_MAJOR_VERSION	Dataset Format Major Version number	NA	NA	*U-INTEGER*	5	38	1005
FORMAT_MINOR_VERSION	Dataset Format Minor Version number	NA	NA	*U-INTEGER*	5	38	1043
PROCESSING_TIME_START	UTC time of the processing at start of processing for the product	NA	NA	*GENERAL TIME*	15	48	1081
PROCESSING_TIME_END	UTC time of the processing at end of processing for the product	NA	NA	*GENERAL TIME*	15	48	1129
PROCESSING_MODE	Identification of the mode of processing	NA	NA	*E-CHAR*	1	34	1177
DISPOSITION_MODE	Identification of the diposition mode	NA	NA	*E-CHAR*	1	34	1211
RECEIVING_GROUND_STATION	Acquisition Station Identification	NA	NA	*E-CHAR*	3	36	1245
RECEIVE_TIME_START	UTC time of the reception at CDA for first Data Item	NA	NA	*GENERAL TIME*	15	48	1281
RECEIVE_TIME_END	UTC time of the reception at CDA for last Data Item	NA	NA	*GENERAL TIME*	15	48	1329
ORBIT_START	Start Orbit Number, counted incrementally since launch. Determined at time of SENSING_START_THEORETICAL	NA	NA	*U-INTEGER*	5	38	1377
ORBIT_END	Stop Orbit Number. Determined at time of SENSING_END_THEORETICAL	NA	NA	*U-INTEGER*	5	38	1415
ACTUAL_PRODUCT_SIZE	Size of the complete product	NA	bytes	*U-INTEGER*	11	44	1453
ASCENDING NODE ORBIT PARAMETERS							
STATE_VECTOR_TIME	Epoch time (in UTC) of the orbital elements and the orbit state vector. This corresponds to the time of crossing the ascending node for ORBIT_START	NA	UTC	*LONG GENERAL TIME*	18	51	1497
SEMI_MAJOR_AXIS	Semi major axis of orbit at time of the ascending node crossing [TRUE-OF-DATE]	NA	mm	*INTEGER*	11	44	1548
ECCENTRICITY	Orbit eccentricity at time of the ascending node crossing [TRUE-OF-DATE]	6	NA	*INTEGER*	11	44	1592

INCLINATION	Orbit inclination at time of the ascending node crossing [TRUE-OF-DATE]	3	deg	*INTEGER*	11	44	1636
PERIGEE_ARGUMENT	Argument of perigee at time of the ascending node crossing [TRUE-OF-DATE]	3	deg	*INTEGER*	11	44	1680
RIGHT_ASCENSION	Right ascension at time of the ascending node crossing [TRUE-OF-DATE]	3	deg	*INTEGER*	11	44	1724
MEAN_ANOMALY	Mean anomaly at time of the ascending node crossing [TRUE-OF-DATE]	3	deg	*INTEGER*	11	44	1768
X_POSITION	X position of the orbit state vector in the orbit frame at ascending node [EARTH-FIXED]	3	m	*INTEGER*	11	44	1812
Y_POSITION	Y position of the orbit state vector in the orbit frame at ascending node [EARTH-FIXED]	3	m	*INTEGER*	11	44	1856
Z_POSITION	Z position of the orbit state vector in the orbit frame at ascending node [EARTH-FIXED]	3	m	*INTEGER*	11	44	1900
X_VELOCITY	X velocity of the orbit state vector in the orbit frame at ascending node [EARTH-FIXED]	3	m/s	*INTEGER*	11	44	1944
Y_VELOCITY	Y velocity of the orbit state vector in the orbit frame at ascending node [EARTH-FIXED]	3	m/s	*INTEGER*	11	44	1988
Z_VELOCITY	Z velocity of the orbit state vector in the orbit frame at ascending node [EARTH-FIXED]	3	m/s	*INTEGER*	11	44	2032
EARTH_SUN_DISTANCE_RATIO	Earth-Sun distance ratio - ratio of current Earth-Sun distance to the Mean Earth-Sun distance which is 1Astronomical Unit or AU. 1 AU has a value of 1.495 978 706 91 x 10^11 m as defined by the "International System of Units", Bureau International des Poids et Mesures	6	NA	*INTEGER*	11	44	2076
LOCATION_TOLERANCE_RADIAL	Nadir Earth location tolerance radial	NA	m	*INTEGER*	11	44	2120
LOCATION_TOLERANCE_CROSSTRACK	Nadir Earth location tolerance cross-track	NA	m	*INTEGER*	11	44	2164
LOCATION_TOLERANCE_ALONGTRACK	Nadir Earth location tolerance along-track	NA	m	*INTEGER*	11	44	2208
YAW_ERROR	Constant Yaw attitude error	3	deg	*INTEGER*	11	44	2252
ROLL_ERROR	Constant Roll attitude error	3	deg	*INTEGER*	11	44	2296
PITCH_ERROR	Constant Pitch attitude error	3	deg	*INTEGER*	11	44	2340
LOCATION SUMMARY							
SUBSAT_LATITUDE_START	Latitude of sub-satellite point at start of the data set	3	Deg	*INTEGER*	11	44	2384
SUBSAT_LONGITUDE_START	Longitude of sub-satellite point at start of the data set	3	Deg	*INTEGER*	11	44	2428
SUBSAT_LATITUDE_END	Latitude of sub-satellite point at end of the data set	3	Deg	*INTEGER*	11	44	2472
SUBSAT_LONGITUDE_END	Longitude of sub-satellite point at end of the data set	3	Deg	*INTEGER*	11	44	2516
Leap Second Information							
LEAP_SECOND	Occurence of Leap second within the product. Field is set to -1, 0 or +1 dependent upon occurrence of leap second and direction.	NA	NA	*INTEGER*	2	35	2560
LEAP_SECOND_UTC	UTC time of occurrence of the Leap Second (If no leap second in the product, value is null)	NA	NA	*GENERAL TIME*	15	48	2595
Record counts							
TOTAL_RECORDS	Total count of all records in the product	NA	NA	*U-INTEGER*	6	39	2643
TOTAL_MPHR	Total count of all MPHRs in product (should always be 1!)	NA	NA	*U-INTEGER*	6	39	2682
TOTAL_SPHR	Total count of all SPHRs in product (should be 0 or 1 only)	NA	NA	*U-INTEGER*	6	39	2721
TOTAL_IPR	Total count of all IPRs in the product	NA	NA	*U-INTEGER*	6	39	2760
TOTAL_GEADR	Total count of all GEADRs in the product	NA	NA	*U-INTEGER*	6	39	2799
TOTAL_GIADR	Total count of all GIADRs in the product	NA	NA	*U-INTEGER*	6	39	2838
TOTAL_VEADR	Total count of all VEADRs in the product	NA	NA	*U-INTEGER*	6	39	2877
TOTAL_VIADR	Total count of all VIADRs in the product	NA	NA	*U-INTEGER*	6	39	2916

TOTAL_MDR	Total count of all MDRs in the product	NA	NA	*U-INTEGER*	6	39	2955
Record Based Generic Quality Flags							
COUNT_DEGRADED_INST_MDR	Count of MDRs with degradation due to instrument problems	NA	NA	*U-INTEGER*	6	39	2994
COUNT_DEGRADED_PROC_MDR	Count of MDRs with degradation due to processing problems	NA	NA	*U-INTEGER*	6	39	3033
COUNT_DEGRADED_INST_MDR_BLOCKS	Count of the number of blocks of MDRs degraded due to degraded instrument	NA	NA	*U-INTEGER*	6	39	3072
COUNT_DEGRADED_PROC_MDR_BLOCKS	Count of the number of blocks of MDRs degraded due to degraded processing	NA	NA	*U-INTEGER*	6	39	3111
Time Based Generic Quality Flags							
DURATION_OF_PRODUCT	The duration of the product in milliseconds	ms	NA	*U-INTEGER*	8	41	3150
MILLISECONDS_OF_DATA_PRESENT	The total amount of data present in the product	ms	NA	*U-INTEGER*	8	41	3191
MILLISECONDS_OF_DATA_MISSING	The total amount of data missing from the prodcut	ms	NA	*U-INTEGER*	8	41	3232
Regional Product Information							
SUBSETTED_PRODUCT	Set when product has been subsetted (e.g. geographically subsetted using a region of interest filter). Implies the presence of one or more UMARF GIADRs in GAD section for product retrieved from UMARF.	NA	NA	*BOOLEAN*	1	34	3273
Size of the Record							3307

Worksheet: IPR

FIELD	DESCRIPTION	SF	UNITS	DIM1	DIM2	DIM3	TYPE	TYPE SIZE	FIELD SIZE	OFFSET
RECORD_HEADER	Generic Record Header	0		1	1	1	REC_HEAD	20	20	0
INTERNAL RECORD POINTER										
POINTER	Pointer to first record of a given class			1	1	1	POINTER	7	7	20
SIZE										27

Worksheet: GEADR

FIELD	DESCRIPTION	SF	UNITS	DIM1	DIM2	DIM3	TYPE	TYPE SIZE	FIELD SIZE	OFFSET
RECORD_HEADER	Generic Record Header	0		1	1	1	REC_HEAD	20	20	0
Product Details										
AUX_DATA_POINTER	Unique pointer to auxiliary dataset.	0		1	1	1	char(100)	100	100	20
TOTAL SIZE										120

Worksheet: VEADR

FIELD	DESCRIPTION	SF	UNITS	DIM1	DIM2	DIM3	TYPE	TYPE SIZE	FIELD SIZE	OFFSET
RECORD_HEADER	Generic Record Header	0		1	1	1	REC_HEAD	20	20	0
Product Details										
AUX_DATA_POINTER	Unique pointer to auxiliary dataset.	0		1	1	1	char(100)	100	100	20
TOTAL SIZE										120

FIELD	DESCRIPTION	SF	UNITS	DIM1	DIM2	DIM3	TYPE	TYPE SIZE	FIELD SIZE	OFFSET
RECORD_HEADER	Generic Record Header	0		1	1	1	REC_HEAD	20	20	0
Status Flag										
STATUS_FLAG	See Table 42	0	N/A	1	1	1	enumerated	1	1	20
Size of the Record	**Size of the Record**									21

No Record Defined

FIELD	DESCRIPTION	SF	UNITS	DIM1	DIM2	DIM3	TYPE	TYPE SIZE	FIELD SIZE	OFFSET
RECORD_HEADER	Generic Record Header	0		1	1	1	REC_HEAD	20	20	0
METOP OBT to UTC Correlation Parameters										
UTC_0	The UTC at the time when the onboard counter has the value of field CCU_OBT_0. Composed of 8 bytes. The first two bytes are the number of days passed since 01-Jan-2000/00:00 UTC. The lsb represents 1 day. The max. value is 65535 (equivalent to 180 years) . The next 4 bytes contain the number of milliseconds for the day. The lsb represents 1 ms and the range is 0 to 86399999. The last 2 bytes contain the number of microseconds in the current millisecond. The lsb represents 1 microsecond and the range is 0 to 999. The sum of the milliseconds and microseconds represents the time of the day.	NA	NA	4	1	1	bitst(16)	2	8	20
CCU_OBT_0	Value of the onboard clock at the time of in field UTC_0. Composed of 6 bytes. The CCU_OBT count is produced by a 256 Hz counter with a 32 bit range. Therefore the lsb which represents one count corresponds to approximately 2^-8 s. It ranges from 0 to 4294967295. That's why the first 2 bytes will always show zero	NA	NA	3	1	1	bitst(16)	2	6	28
CLOCK_STEP	Gradient of the time correlation function. Corresponds to the time in picoseconds for one increment of the on-board clock (CCU_OBT). Composed of 4 bytes. The lsb is in units of 10^-12 s per count, whereas a count is approximately 2^-8 s. UTC = CLOCK_STEP * (ISP_OBT/256 – CCU_OBT_0) + UTC_0 CCU_OBT count is produced by a 2^8 Hz counter. ISP_OBT count is produced by a 2^16 Hz counter. Therefore it has to be divided by 256 to match the units of CLOCK_STEP (as shown in the above equation).	NA	NA	2	1	1	bitst(16)	2	4	34
Size of the Record										38

FIELD	DESCRIPTION	SF	UNITS	DIM1	DIM2	DIM3	TYPE	TYPE SIZE	FIELD SIZE	OFFSET
RECORD_HEADER	Generic Record Header	0		1	1	1	REC_HEAD	20	20	0
GENERIC QUALITY INDICATORS										
DEGRADED_INST_MDR	Quality of MDR has been degraded from nominal due to an instrument degradation.	NA	NA	1	1	1	boolean	1	1	20
DEGRADED_PROC_MDR	Quality of MDR has been degraded from nominal due to a processing degradation.(Currently not used. Default to a value of FALSE)	NA	NA	1	1	1	boolean	1	1	21
Instrument Data										
SIZE_INST_DATA	Size of the INST_DATA array (**n)**	0	bytes	1	1	1	u-integer4	4	4	22
INST_DATA	Instrument or satellite source packet data or GAC/SAIP/STIP data as stripped from the telemetry stream	0	0	**n**	1	1	byte	1	depends on instrument and satellite	26
Size of the Record										depends on instrument and satellite

Field Type	Size in Bytes
bitst(16)	2
bitst(24)	3
bitst(32)	4
bitst(8)	1
boolean	1
byte	1
char(1)	1
char(100)	100
char(108)	108
char(2)	2
char(3)	3
char(4)	4
e-char(1)	1
e-char(2)	2
e-char(3)	3
e-char(4)	4
enumerated	1
general time	15
long general time	18
integer2	2
integer4	4
integer8	8
long cds time	8
POINTER	7
REC_HEAD	20
short cds time	6
u-byte	1
u-integer2	2
u-integer4	4
u-integer8	8

NOTE: Table must be sorted into ascending order