Customer	**:**	**ESRIN**	**Document Ref**	**:**	**IDEAS+-SER-IPF-SPE-2337**
Contract No	**:**	**4000111304/14/I-AM**	**Issue Date**	**:**	**31 August 2015**
ESA Doc Ref	**:**	**PO-RS-MDA-GS-2009**	**Issue**	**:**	**3 / G**

TITLE: ENVISAT-1 PRODUCTS SPECIFICATIONS

ANNEX A: PRODUCT DATA CONVENTIONS

Abstract : This document contains data conventions used by the other volumes

Written by : Massimo Cardaci Digitally signed by Massimo Cardaci DN: email=Massimo.Cardaci@serco.com,cn=Massimo Cardaci,o=SERCO,c=IT Date: 2015.10.12 14:55:34 +0200 Signed with DigiSigner Free!

M. Cardaci- IDEAS+

Approval : Lidia Saavedra de Miguel Digitally signed by Lidia Saavedra de Miguel DN: email=Lidia.Saavedrademiguel@serco.com,cn=Lidia Saavedra de Miguel,o=SERCO,c=IT Date: 2015.10.12 15:02:18 +0200 Signed with DigiSigner Free!

Lidia Saavedra de Miguel – IDEAS+

Accepted : Bojan Bojkov Digitally signed by Bojan Bojkov DN: email=bojan.bojkov@esa.int,cn=Bojan Bojkov,c=IT Date: 2015.10.12 14:55:46 +0200 Signed with DigiSigner Free!

Bojan Bojkov - ESA

Distribution :

Hard Copy File:
Filename: IDEAS+-SER-IPF-SPE-2337_3 / G.doc

TABLE OF CONTENTS

INDEX OF TABLES

INDEX OF FIGURES

No figures in the document

AMENDMENT POLICY

This document shall be amended by releasing a new edition of the document in its entirety. The Amendment Record Sheet below records the history and issue status of this document.

AMENDMENT RECORD SHEET

ISSUE	REVISION	DATE	CHANGE STATUS	ORIGIN
1	A	12/01/1996	Issue 1	SRR
1	B	16/02/1996	SCR #16, CR #16 Issue 1, Revision B Reason for Change: Updated to reflect information in PO-TN-ESA-GS-0381 and to address RIDs of Feb. 2/96 pertaining to the Level 0 structure. MPH, SPH, DSD, and DSR structures modified. Table added showing generalized Level 0 product structure. RIDs Addressed: ESA/0001: FEP header defined ESA/0002: PF-Host time stamp clarified ESA/0004: Processing PCD added ESA/0006: AF PCD ADS and DSD added ESA/0007: page A-3 updated ESA/0008: page B-3 updated ESA/0009: Table 8.1.1 modified ESA/0011: TBD changed to Range/ Doppler ESA/0013: FEP header defined ESA/0014: Table 8.4.7.4-2 corrected CSF/1:	SRR

ISSUE	REVISION	DATE	CHANGE STATUS	ORIGIN
			filename in MPH corrected CSF/2: page A-3 updated CSF/3: MPH PCD information updated CSF/5: DSD added to Level 0 SPH CSF/6: Section on AATSR updated and re-issued CSF/8: AATSR_O Summary Sheet updated	
1	C	04/04/1996	SCR #38, CR #38 Issue 1, Revision C Reason for Change: Updated Sections 1-6, 17 and Annex A to reflect changes discussed at the Products Review Meeting #1, March 5-8, 1996, as per action item “AI MDA 6 April 96” from PO-MN-ESA-00416, Pg. 35.	Products Review Meeting #1
2	A	20/05/1996	SCR #71, CR #71 Issue 2 Separate volume created	
2	B	10/02/1997	SCR #102, CR #102 Issue 2, Revision B Reason for Change: Originator_ID codes created. Minor updates.	Products Review Meeting #2
3	A	27/05/1997	SCR #169 , CR #169 Issue 3, Revision A Bit numbering convention corrected MJD format placed in table	Products Review Meeting #3
3	B	20/09/2001	Definition of Ac as recovering both unsigned and signed char types.	
3	C	10/09/2002	Added IECF as Originator ID on table A.1-1	
3	D	05/05/2004	Added ACRI and FINPAC as Originator IDs on table A.1-1 and updated the complete list to reflect operational status	P.Gilles mail of 5/ 05/2004 Subj: Prod

ISSUE	REVISION	DATE	CHANGE STATUS	ORIGIN
				Spec Update
3	E	20/06/2005	New originator IDs: PAM for matera NRT production O_M for Orbite Mission	Request from P.F emenias and Change of Scenario on Matera
3	F	09/01/2014	Update of Originators list to include DSI	JCCB-CP-323 - UPDATE OF ENVISAT PRODUCT SPECIFICATIONS for DSI
			Update of interpretation of product counter in case of reprocessing campaigns	JCCB-CP-323 - UPDATE OF ENVISAT PRODUCT SPECIFICATIONS for DSI
3	G	31/08/2015	Update of Stations list to reference external ESA document (“EO Parameter Code List” - PGSI-GSEG-EOPG-TN-07-0001)	JCCB-CP-323 - update of envisat product specifications for dsi (updated decision JCCB 20.08.2015)

REGISTER OF CHANGES

Section	Change
Table A1--1	Added “DSI” centre in field “Originator ID”
Table A1--1	Added reference to “note 1” in field “counter” (reference in the table and text after the table.
All	Porting of template
All	Updated with IDEAS+ contract details
Table A1--1	Field #3 (originator ID): replaced table with reference to ESA document: “EO Parameter Code List” (PGSI-GSEG-EOPG-TN-07-0001). Existing list replaced with footnote
Table A1--1	Field #3 (originator ID): added LRC = LRAC backup chain

ANNEX A PRODUCT DATA CONVENTIONS

This appendix summarizes the product conventions used in this document

A.1 PRODUCT FILE NAMING

The first field of the Main Product Header contains the product name. The naming convention for products is described below.

```
filename = <product_ID> <processing_stage_flag>

<originator_ID><start_day> <"_"> <start_time> <"_"> <duration> <phase>
<cycle> <"_"> <relative_orbit> <"_"> <absolute_orbit> <"_"><counter>
<"."> <satellite_ID> <.extension>
```

The naming convention for auxiliary data files is described in Volume 16.

Table A.1-1 Product Name Fields

Field.	Size in Characters	Description
Product_ID	10	10 character string identifies sensor, mode and processing level. See Volume 4 for details. Characters not used are replaced with an underscore character.
Processing Stage flag	1	Set to "N" for Near Real Time product Set to "V" for fully validated (consolidated) product Set to "T" for Test product Set to "S" for a special product. Letters between N and V are assigned in order of level of consolidation (i.e., closer to V = better consolidated)
originator ID	3	Identification of the center which generated the file. The 3 character code may be one of the codes listed in **NOTE (2)** below. All codes are TBC by ESA
start_day	8	In the case of instrument products it corresponds to the start day of the product from the UTC time of the first DSR. The format is YYYYMMDD. For Auxiliary files it may correspond to file creation date.
start_time	6	Time coverage of the product expressed in seconds. If the duration of a product is not

Table A.1-1 Product Name Fields		
Field.	**Size in Characters**	**Description**
		relevant information it will be set to "00000000".
duration	8	Time coverage of the product expressed in seconds. If the duration of a product is not relevant information it will be set to "00000000".
phase	1	Mission phase identifier
cycle	3	Cycle number within the mission phase
relative_orbit	5	Relative orbit number within the cycle at the beginning of the product
absolute_orbit	5	Absolute orbit at the beginning of the product
counter	4	Numerical wrap-around counter for quick file identification. For a given product type the counter is incremented by 1 for each new product generated by the product originator (see note 1)
satellite ID	2	E1 = ERS-1, E2 = ERS-2, N1 = ENVISAT-1
Extension	Variable	Optional field. Used only for distribution to users to indicate common archiving and compression standards if used (e.g., .gz, .Z, .tar, .tarZ, .gif, .jpeg, etc.)

Note:

(1) In the case of Reprocessing Campaigns, the 4 digits of the file counter could optionally be set to the same value (as an example: "0000"). The file counter will in this case only differentiate (i.e. be incremented) in case of real processing duplicates.

(2) At the time of this PSS update, the **current list of stations** is attached here. See also The up to date list is available from the following ESA document: "EO Parameter Code List" (PGSI-GSEG-EOPG-TN-07-0001), for eventual subsequent additions:

Code => Station
PDK => PDHS-K
PDE => PDHS-E
IEC => IECF
LRA => LRAC
LRC => LRAC (backup chain)
PDC => PDCC
FOS => FOS-ES
PDA => PDAS-F
PAM => Matera for NRT production
UPA => UK-PAC

DPA => D-PAC
IPA => I-PAC
FPA => F-PAC
SPA => S-PAC
EPA => E-PAC
ECM => ECMWF
ACR => ACRI
FIN => FINPAC
O_M => Orbite Mission
DSI => DSI

For example, a fully consolidated Level 0 MIPAS product which contains data starting on Feb 10, 1999 at 13:32:54 covering a complete orbit (6040 seconds), from data acquired during mission phase “A”, cycle 31, relative orbit 67, absolute orbit 15598, generated at the D-PAC and compressed using the gzip utility would have the form:

```
MIP_NL__0PVD-P19990210_133254_00006040A031_00067_15598_0324.N1.gz
```

This file naming convention assumes the use of an operating system that allows long filenames. Platforms which use operating systems that do not support long filenames must use a subdirectory tree. The maximum length of a subdirectory name is eight characters.
For example, an MS-DOS file system (name limited to 12 characters with a periodon the ninth) would use a subdirectory structured as:

```
<first 8 characters of Product ID> \ <last 2 characters of
Product_ID><Processing_Stage_Flag> <originator_ID> \ <start_day> \
<start_time> \ <duration> \ <phase> <cycle> \ <relative_orbit> \
<absolute_orbit> \ <counter> <satellite_ID><.extension>

e.g

MIP_NL__\0PVD-P\19990210\133254\00006040\A031\00067\15598\0324N1.gz
```

A.2 DATA REPRESENTATION

The eligible data types for product structures are listed in Table A.2-1.

Table A.2-1 Data Types

Variable Type	C Type	Abbreviation	Range
Character	char	sc: signed char	-128 to 127 (2's comp.)
		uc: unsigned char	0 to 255
2-byte integer	short	ss: signed short integer	-32768 to 32767 (2's comp)
		us: unsigned short integer	0 to 65535
4-byte integer	long	sl: signed long integer	-2147483648 to 2147483647
		ul: unsigned long integer	0 to 4294967295
8-byte integer	long long	sd: signed long long integer	-9223372036854775808 to 9223372036854775807
		ud: unsigned long long integer	0 to 18446744073709551615
4-byte single precision floating point	float	fl	3.4028e+38 (max) 1.17549e-38 (min)
8-byte double precision floating point	double	do	1.79e+308 (max) 2.22e-308 (min)

The IEEE 754-1985 is the chosen standard for storing real numbers.

For header structures which use ASCII values, the following methods for representing binary data types in ASCII are followed:

Table A.2-2 ASCII Equivalent Formats

Variable Type	C Type	ASCII format	ASCII Abbreviation
Character	sc: signed char	Single ASCII character (if designated a 1 byte number in original MPH or SPH format will be SXXX[a]	Ac
	uc: unsigned char		
2-byte integer	ss: signed short integer	SXXXXX	As

Table A.2-2 ASCII Equivalent Formats

Variable Type	C Type	ASCII format	ASCII Abbreviation
	us: unsigned short integer	(6 bytes)	
4-byte integer	sl: signed long integer ul: unsigned long integer	SXXXXXXXXXX (11 bytes)	Al
8-byte integer	sd: signed long long integer ud: unsigned long long integer	SXXXXXXXXXXXXXXXXXXXX (21 bytes)	Ad
4-byte single precision floating point	fl	SX.XXXXXXXXESXX (15 bytes)	Afl
8-byte double precision floating point	do	SX.XXXXXXXXXXXXXXXXXESXXX (25 bytes) S.XXXXXX (8 bytes) SXXXX.XXXXXX (12 bytes) SXXXXXXX.XXX (12 bytes)	Ado Ado06 Ado46 Ado73

a. S = sign (+ or -), X = a single number in ASCII format between 0 and 9

Note that the sign is always included, even for positive numbers, and unused positions are set to zero. E.g. the number 1.435E12 is represented as +1.43500000E+12; the long integer 123456789 is represented as +0123456789.
The range of value allowed for each ASCII format can be deduced of the format, e.g. Ac range is -999 ... 999 . However this does not means that all the representable values will be used, e.g. some field counter declared as Ac will never take negative values.

A.2.1 ASCII Character Set

The standard ASCII character code set used for ENVISAT Products is the first 128 characters of the 8-bit ISO8859 - 1 character code, which is identical to the long established US-ASCII 7-bit character code. For the sake of clarity, the complete list of ASCII codes used for products is given below. The rules used to create ASCII header structures are given in Volume 5 of this document. When ASCII character strings are included in binary data sets, the string is left-justified within the field.

ASCII blank-space characters are added to the right of the string to fill the field.

Note the symbol Ø is used in the documentation to indicate the position of an ASCII blank-space character (character 32) in Table A.2.1-1.

Table A.2.1-1 Decimal Value and corresponding ASCII character

0 NUL	1 SOH	2 STX	3 ETX	4 EOT	5 ENQ	6 ACK	7 BEL
8 BS	9 HT	10 NL	11 VT	12 NP	13 CR	14 SO	15 SI
16 DLE	17 DC1	18 DC2	19 DC3	20 DC4	21 NAK	22 SYN	23 ETB
24 CAN	25 EM	26 SUB	27 ESC	28 FS	29 GS	30 RS	31 US
32 SP	33 !	34 "	35 #	36 $	37 %	38 &	39 '
40 (	41)	42 *	43 +	44 ,	45 -	46 .	47 /
48 0	49 1	50 2	51 3	52 4	53 5	54 6	55 7
56 8	57 9	58 :	59 ;	60 <	61 =	62 >	63 ?
64 @	65 A	66 B	67 C	68 D	69 E	70 F	71 G

72 H	73 I	74 J	75 K	76 L	77 M	78 N	79 O
80 P	81 Q	82 R	83 S	84 T	85 U	86 V	87 W
88 X	89 Y	90 Z	91 [	92 \	93]	94 ^	95 _
96 ‘	97 a	98 b	99 c	100 d	101 e	102 f	103 g
104 h	105 i	106 j	107 k	108 l	109 m	110 n	111 o
112 p	113 q	114 r	115 s	116 t	117 u	118 v	119 w
120 x	121 y	122 z	123 {	124 |	125 }	126 ~	127 DEL

A.2.2 Logical Values

Logical values are values which may be either true or false. The following convention is followed:

Table A.2.2-1 Logical Values

Logical	Value	ASCII Representation [a]
True	1	ASCII code 49
False	0	ASCII code 48

a. See Table A.2.1-1

A.2.3 Unused Fields

In cases where a field is not fully filled by the value which it contains, placeholder values are used. For ASCII strings, the placeholder character is the ASCII blank-space character (ASCII character 32). For numerical-values, the placeholder value is zero unless otherwise stated. For ASCII numerics (defined in Table A.2-2 above) an ASCII numeric of 0 (in the appropriate format) may be used if specified.

A.3 BIT / BYTE NUMBERING

For the purpose of identifying bits within a multi-byte structure, the numbering convention shown below is used. Byte 0 is the most significant byte. It is transmitted before byte 1. Within a byte, bit 0 is the least significant bit. This is the convention defined in the Product Format Guidelines (document R-1).

1 byte structure:

Bytes	BYTE 0							
Bits	7	6	5	4	3	2	1	0

2 byte structure:

Bytes	BYTE 0								BYTE 1							
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

3 byte structure:

Bytes	BYTE 0								BYTE 1								BYTE 2							
Bits	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

4 byte structure:

Bytes	BYTE 0								BYTE 1								BYTE 2								BYTE 3							
Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

A.4 TIME

Within the PDS time is used with an accuracy of 1 microsecond, expressed as:

- **UTC** (Universal Time Coordinate) almost equivalent to GMT (Greenwich Meridian Time) presented as a string of 27 significant characters with the format:

```
DD-MMM-YYYYØhh:mm:ss.tttttt
```

where

DD:	day	[1..31]
MMM:	month	[JAN, FEB,...NOV, DEC]
YYYY:	year	[1950..2050]

Ø: blank character
hh: hour [00..23]
mm: minutes [00..59]
ss: second [00..59]
tttttt: us [000000..999999] may be blanked by spaces if irrelevant

e.g. December 29, 1999 at 10:00 is coded as
29-DEC-1999 10:00:00.000000 or 29-DEC-1999 10:00:00

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

Table A.4-1 MJD format

N	Description	Units	Byte Length	Data Type	Dim
1	Number of days elapsed since the 1st of January 2000 at 0:0 hour. It may be negative, and is thus a signed long integer	days	4	sl	1
2	Number of seconds elapsed since the beginning of that day	s	4	ul	1
3	Number of microseconds elapsed since the last second	us	4	ul	1
Total			**12**		

e.g., December 29, 1999 at 10:00 is coded as
{-3 , 36000, 0}

As a general rule, UTC time format is used in the MPH and SPH, while MJD format will be used when time stamps are required for DSRs within a DS.

A.5 GEOLOCATION INFORMATION

The WGS84 co-ordinate system is used for all latitude/longitude geolocation. The system is described in detail in Document R-20.
Geolocation information is expressed within ENVISAT products using the following convention:

latitude:	4 byte signed long integer units = 10^{-6} degrees positive north (-90 = south pole, +90 = north pole)

longitude:	4 byte signed long integer units = 10^{-6} degrees positive east, 0 = Greenwich meridian, range: [-180, 180) i.e., west direction includes -180, east does not include +180

Latitude is always listed prior to longitude.

A.6 BUFR AND GRIB FORMAT

The Binary Universal Form for the Representation of meteorological data (BUFR) is a bit-oriented data exchange format used in meteorology. It is not supported within the ENVISAT PDS but ENVISAT products may be converted to this format outside the PDS. The format is described in Document R-21.

GRIB is the GRIdded Binary form for meteorological data representation. It is not supported within the ENVISAT PDS, but auxiliary data accepted into the PDS from the ECMWF may be in this format. The format is described in Documents R-28 and R-29.

A.7 SIZES

All sizes provided in this document follow the following convention:

- 1 kilobyte = 1x10 bytes = 1 kB or 1 kByte
- 1 megabyte = 1x10 bytes = 1 MB or 1 MByte

A.8 ALIGNMENT IN STRUCTURES FOR THE IBM SP2

All sizes listed in the Product Specifications assume byte aligned structures. However, the IBM SP2 aligns structures in memory according to the table below.

Table A.8-1 Type Size and Alignment for the RISC System/6000

Type	Alignment of Member	Size (Bytes)
char	byte aligned	1
short	2-byte aligned	2

Table A.8-1 Type Size and Alignment for the RISC System/6000

Type	Alignment of Member	Size (Bytes)
(long) int	4-byte aligned	4
long long int	8-byte aligned	8
pointer	4-byte aligned	4
float	4-byte aligned	4
double	8-byte aligned if -qalign=natural. Otherwise, word aligned.	8
long double with -qlongdouble or -qldb1128 option.	16-byte aligned if -qalign=natural. Otherwise, word aligned.	16

This means that if data is stored as structures, the sizes listed in the Product Specifications may not correspond exactly to the size of memory the IBM SP2allocates to store them.
For example, suppose an ADSR consisted of 5 characters followed by a float:

e.g.

```
unsigned char data1[5];
float data2;
```

Since a float is 4 bytes and a char is 1 byte, the size of this data would be listed as 9 bytes.
However, suppose this data was declared as a structure (as in the DDT):

e.g.

```
struct st
{
unsigned char data1[5];
float data2;
};
```

According to Table A.8-1, the IBM will only store a float in memory beginning on a 4-byte boundary. Therefore, it will add 3 bytes of padding to the unsigned character array before storing the float. Thus, the actual size of the structure in memory becomes 5 bytes + 3 bytes padding + 4 byte float = 12 bytes.

Obviously, this padding is not desirable as it tends to bloat the size of products and causes the byte alignment to differ from that of the Product Specifications. Output products must have the same size and alignment as specified within this document.

There are two possible solutions to this problem:

1. The IBM C++ compiler has a flag which can be set to force the use of byte aligned structures.
2. Elements of a structure may be copied individually to ensure proper alignment and size of data members.

DISTRIBUTION LIST

NAME	COPY	NAME	COPY
Henri Laur (ESA)	1		
Bojan Bojkov (ESA)	1		
Simon Jutz (ESA)	1		
Pascal Gilles (ESA)	1		

End of Document