SMOS L1 Processor Prototype 3.4.0

Software Release Note

Code	**:**	SO-SRN-DME-L1PP-0237
Issue	**:**	1.0
Date	**:**	31/05/10

	Name	Function	Signature
Prepared by	J. Barbosa	Project Engineer	
	A. Gutiérrez	Project Engineer	
	J. Freitas	Project Engineer	
Checked by	J.Freitas	Quality A. Manager	
Approved by	J. Barbosa	Project Manager	

DEIMOS Engenharia
Av. D. João II, Lote 1.17, Torre Zen, 10º
1998-023 Lisboa, PORTUGAL
Tel: +351 21 893 3017
Fax: +351 21 896 9099
E-mail: mailto:deimos@deimos.com.pt

This page intentionally left blank

Document Information

Contract Data	Classification
Contract Number: DE04/B-434/P	**Internal** ☐ **Public** ☑
Contract Issuer: EADS CASA Espacio	**Industry** ☐ **Confidential** ☐

Internal Distribution		
Name	Unit	Copies

External Distribution		
Name	Organisation	Copies
Josep Closa	EADS CASA Espacio	1
Michele Zundo	ESA	1

Archiving	
Word Processor:	MS Word 2000
File Name:	SO-SRN-DME-L1PP-0211-L1PP-3-2-1-Software-Release-Note.doc
Archive Code:	P/SUM/DME/03/013-039

Document Status Log

Issue	Change description	Date	Approved
1.0	Phase 5 V3.4.0 L1PP Release	2010-05-31	

Table of Contents

List of Tables

1. INTRODUCTION

1.1. Purpose and Scope

This document provides the Software Release Note for the SMOS L1 Processor Prototype v3.4.0 and associated tools. It also details the Installation and Compilation procedures.

1.2. Acronyms and Abbreviations

CFI	Customer Furnished Item
COTS	Commercial Off-The-Shelf
EE	Earth Explorer
EEFH	Earth Explorer File Handling CFI (ASCII XML library)
GUI	Graphical User Interface
HTML	HyperText Markup Language
L1PP	Level 1 processor prototype
SMOS	Soil Moisture and Ocean Salinity
SVP	Software Validation Plan
XML	Extended Markup Language
XSL	eXtensible Stylesheet Language

Table 1: Table of Acronyms.

For the complete list of acronyms, please refer to the document SO-LI-CASA-PLM-0094 “Directory of Acronyms and abbreviations”.

1.3. Applicable and Reference Documents

1.3.1. Applicable Documents

Ref.	Code	Title	Issue
AD.1	SO-SOW-CASA-PLM-0855	Level 1 Processor Prototype Development Phase 3 and Support and Analysis Activities. Statement of Work	01
AD.2	SO-UM-DME-L1PP-0005	EE XML/Binary CFI File Handling Library	2.1
AD.3	SO-DS-DME-L1PP-0007	SMOS L1 Processor L0 to L1a Data Processing Model	2.12
AD.4	SO-DS-DME-L1PP-0012	SMOS L1 Processor Prototype ADD	1.3
AD.5	BinX	Editkt::BinX 1.2 Developer's Guide	1.2
AD.6	SO-TR-DME-L1PP-0018	System Validation Plan	2.3
AD.7	SO-TN-IDR-GS-0005	SMOS L1 DPGS Product Specification	5.16
AD.8	SO-DS-DME-L1PP-0008	SMOS L1 Processor L1a to L1b Data Processing Model	2.11
AD.9	SO-DS-DME-L1PP-0009	SMOS L1 Processor L1c Data Processing Model	2.7
AD.10	SO-DS-DME-L1PP-0011	SMOS L1 Processor Algorithm Theoretical Baseline Definition	2.8
AD.11	SO-UM-DME-L1PP-0016	SMOS L1 Processor Prototype User Manual	2.14

Table 2: Applicable Documents.

1.3.2. Reference Documents

Ref.	Code	Title	Issue
RD 1	CS-MA-DMS-GS-0001	Earth Explorer Mission CFI Software MISSION CONVENTIONS DOCUMENT	1.3
RD 2	PE-TN-ESA-GS-0001	Earth Explorer Ground Segment File Format Standard	1.4
RD 3	CS-MA-DMS-GS-0002	Earth Explorer Mission CFI Software GENERAL SOFTWARE USER MANUAL	3.7.2
RD 4	SO-LI-CASA-PLM-0094	Directory of Acronyms and abbreviations	

Table 3: Reference Documents

2. Version Description

2.1. Baseline

This is version 3.4.0 of the SMOS L1 Processor Prototype.

2.2. Changes Implemented

This is the fifth and final release of the SMOS L1 Processor Prototype Phase 5, corresponding to the build version 3.4.0. The following changes with respect to the previous release (L1PP Build 3.3.0) are included in this release.

L1PP Version: 3.4.0

CVS Tag: SMOS-P5-REL-L1PP-3_4_0-20100531

Creation Date: 2010-05-31

Corrected the following SPRs with respect to the previous release (v 3.3.0):

- SPR-533 - Use of the copolar flag to generate strictly copolar G matrices
- SPR 532 - Implement experimental integration times in full pol into theoretical equation for Radiometric Accuracy in L1c
- SPR 531 - Faraday Rotation equation sign mismatch between L2 and L1
- SPR 530 - ESA/CASA request that ANIR orquestration be moved before CRSD
- SPR 527 - FWF shape averaging is not considering the minimum number of subevents configuration
- SPR 526 - SPAR CAS correction factors need to be applied per path from NS to LICEF
- SPR 525 - Trec sensitivities used to propagate NIR measurements are defined at a different plane
- SPR 524 - L1PP is not able to read L1 OPER files when configured to work in OPER mode and TEST ADF
- SPR 523 - Data Provider loses track of ANIR files if input directory is different from output directory
- SPR 522 - Incorrect arm assignment on Hub Noise Source when mixing nominal and redundant configurations
- SPR 521 - Incorrect CFI Time correlation initialisation
- SPR 520 - Filtering of RFI events in Unoise calibration

- SPR 514 - Theoretical improvement of Tref application in Gibbs 1/2 casesSPR 518 - Incorrect reference for used ANIR files in SC/TAR_X1A headers

In addition to the SPR correction, the following updates were made:

- Major updates
  - As part of SPR-521 enhancement, L1PP can now read IERS B bulletins, provided they respect the old format that was discontinued in February 2010, and that they are available inside $L1PP_ROOT/config/iers_bulletins/bulletinb.XXX
- Minor updates
  - This L1PP Installer includes the latest XML RW API and DPGS schemas available at the time of creation (v04.07.04)

2.3. Differences wrt Baseline

This version of the L1 Prototype (3.4.0) has been delivered together with the latest documentation baseline of L1 Processing (see Section 1.3). The latest instrument and processing baseline is fully implemented with some exceptions:

- On all DPGS products, the quality information (in both headers and datablocks) is set to zero as L1PP v3.4.0 is not computing any quality indicators (except for those already present in the L0 headers which are simply transmitted onwards)
- The following changes are to be understood as L1PP "evolutions", which are implemented and tested using L1PP framework before issuing formally a recommendation for DPGS implementation (or not). Some of them are already known not to be needed in DPGS, but are reported here for completeness:
  - Configurable option to use new PMS external calibration baseline or the nominal PMS internal calibration (external PMS calibration is not recommended for DPGS)
  - Configurable option to activate Unoise correction for baselines covered only by a common Noise Source (no recommendation has yet been issued)
  - Configurable option to decimate Local Oscillator sequences by configuration (not recommended for DPGS)
  - NIR Consolidation process (recommended, but needs ANIR format update before DPGS implementation)
  - Moved NIR orchestration before CRSx orchestration
  - Implemented usage of IERS bulletin B for time initialisation (recommended for DPGS)

2.4. Implementation of Mandatory Fields

A table reporting the mandatory fields for Level 1c products can be found in Annex. This table only reports the status of the L1C product headers. All datablock values are set according to the specification.

2.5. Known limitations and bugs

As of Phase 5 (L1PP v3.1.0), only the 64bit version of the SW is supported.

The following Software Problem Reports were identified and remain to be corrected:

- SPR 529 - ANIR format update needed in order to perform consolidation in DPGS
- SPR 528 - ESA/CASA request that Tp7 is averaged over each half orbit instead of used instantaneously
- SPR 463 - T4' reconstruction still present some artefacts

For reporting additional SPRs, please use the Problem report form available at the L1PP project's webpage:

http://www.smos.com.pt/problem_report.html.

3. INSTALLATION PROCEDURE

3.1. Hardware Requirements

L1PP may be run on any x86 architecture (PentiumIV, Xenon, AMD Opteron, …) 64 bits with LINUX installed. The memory and disk resources needed for executing the prototype depend on the type of algorithms being used. For full functionality, the user needs the following resources:

- 18,5 GB of Disk Space – from these 18,5GB, 16GB are needed for generating the G-Matrix ADF[1]. After the generation of the ADF, the L1PP needs around 10GB of disk space for all the ADFs, libraries and some available space for products generation;
- Minimum 4GB RAM - the RAM memory available will be critical when handling the G-Matrix ADF (Level 1B). If the user wants to use the G-Matrix ADF for performing the Foreign Sources Correction in Full Polarisation, a minimum of 10GB is advisable (16 GB recommended). If the user has less than 8 GB of RAM, a swap partition of at least 8 GB is also recommended as the total amount of memory needed in L1b processing may reach 10GB for a full pol product half-orbit.

As mentioned before, only the 64bit version of L1PP is maintained, which requires a 64bit HW platform to execute.

3.2. Dependencies

In order to run the SMOS L1PP, it is recommended to have the following software already installed in the target platform:

- GCC 4.3 (http://gcc.gnu.org)
- glibc 2.3.4 (http://gcc.gnu.org)
- gfortran (version included in gcc 4.3)
- Earth Explorer CFI 3.7.2 (which already includes libxml2 2.6.16) (http://eop-cfi.esa.int/)
- BinaryXML File Handler 3.7 (already includes BinX 1.2.6) (http://www.smos.esa.int/)
- xerces 2.8.0 (http://xml.apache.org/xercesc/)
- XML RW API (ftp://193.146.123.163/smos/software/XML_RW_API/)

[1] The ADF occupies around 8GB of disk space. However, since it is not provided with the Install Kit, it must be generated by the prototype (refer to [AD.12] for further details on generating the G-Matrix ADF). For generating it a minimum of 2X8GB of disk space available is needed. After the generation of the ADF, only the 8GB occupied by the ADF are needed.

- xerces 2.7.0 (http://xml.apache.org/xercesc/) (needed by XML RW API)
- DOM4J 1.5.2 (http://www.dom4j.org)
- Jaxen 1.1 (http://jaxen.codehaus.org)
- Swing 1.0
- Log4c 1.2.0 (http://log4c.sourceforge.net)
- Java J2SE SDK 1.5.0 (http://java.sun.com/j2se/1.5.0/download.html)
- Lapack 3.0 (http://www.netlib.org/lapack/)
- Blas 3.0 (version included in Lapack 3.0)
- Fftw 3.1.2 (http://www.fftw.org/)

Each product referred above should be installed according to its own instructions in the target platform.

Nevertheless, the L1PP Install Kit already contains some of the above dynamic libraries in case the user has not installed them locally. The included libraries are:

- BinaryXML 3.7
- xerces 2.8.0 (http://xml.apache.org/xerces2-j/)
- XML RW API (ftp://193.146.123.163/smos/software/XML_RW_API/)
- xerces 2.7.0 (http://xml.apache.org/xercesc/) (needed by XML RW API)
- Libxml2 2.6.16
- Log4c 1.2.0
- DOM4J 1.5.2
- Jaxen 1.1
- Swing 1.0
- Lapack 3.0
- Blas 3.0 (version included in Lapack 3.0)
- Fftw 3.1.2
- gfortran (version included in gcc 4.3)

Libraries *dom4j.jar*, *jaxen-1.1-beta-8.jar* and *swing-layout-1.0.jar* are provided in *l1pp/lib* directory (no extraction or installation required).

The rest of the libraries may need to be compiled as appropriate, so it may be possible that they need to be installed as root. Please check each installation instructions for this purpose, although for GNU packages it can be done simply by executing *./configure* and then *make*.

3.3. Installation Kit Description

The L1 Processor Prototype is provided as an InstallKit composed of a single tgz file:

- *L1PP-Installer.tgz*: contains the installation script, readme and a tgz file with all the libraries, binaries and configuration files of the L1 Prototype. The package contains inside a pre-configured structure of the L1PP working directory with the Linux 64 bits version and Mac OS X of the L1PP. All required configuration files, product schemas and header templates are also included;

In order to run the prototype the user will also need the Auxiliary Data Files and the test data scenarios, which are provided separately:

- *ADFPackage_3_4.tgz*: contains all the Auxiliary Data Files needed by the prototype as well as the corresponding xml schemas and documentation;

- *TestDataPackage_3_4.tgz*: contains several test data scenarios (L0 products) and corresponding configuration files in order to be used directly by the prototype. In addition, this package also includes documents with the test scenarios description and with the product format specification.

3.4. L1PP Installation Steps

In order to install the L1PP, the user shall extract the *tgz* file and execute the install script in the command line:

$ tar –xzvf L1PP-Installer.tgz

$ sh InstallKitL1PP.sh $INSTALL_PATH

Where $INSTALL_PATH is the directory where the user wants to install the prototype. For example, if we want to create it in */home/smos/tools/*, the command to execute shall be:

$ sh InstallAllL1PP.sh /home/smos/tools/

and the script will extract and create a new directory inside *tools* called *l1pp-<version>* with all the L1 Prototype contents (Libraries, configuration files, xml schemas, product headers, etc.).

The installation script shall also update the configuration files being extracted so that the paths referred inside are correctly referenced to the installation directory.

Finally, the install script will attempt to make a soft link to the Ionospheric Models provided with L1PP into the system directory */usr/local/etc/*. If this is not possible because the user does not have privileges, it shall be detected at installation time and the user will be prompted to execute the link command with root privileges. The non-availability of this Ionospheric Models will not stop the L1 Prototype from running; it will simply disable the TEC and IGRF model computation at L1c processing.

After the installation is complete, the directories structure shown in Section 5shall be created:

After installing the prototype, the user will need the Auxiliary Data Files contained in *ADFPackage_3_2.tgz*. After extracting the package, a soft link pointing from *<l1pp home>/data/adf* directory to *ADF-Package-3_2/adf* shall be created.

Finally the user will need to fetch the directories *<l1pp home>/data/l1a-in* , *<l1pp home>/data/l1b-in, <l1pp home>/data/l1c-in* with the products provided as examples on the test data package.

As part of the 3.4.0 delivery, the *dpgs_interface* directory is provided inside the Install Kit. This directory contains the latest status of the DPGS-V3 products and schemas (v04.07.04) at the time of the release.

3.4.1. Environment Variables

For execution of the prototype, a new environment variable named L1PP_ROOT must be created, pointing to the *l1pp* directory. For instance, if the *$INSTALL_PATH* is */home/smos/tools/* , the L1PP will be installed in */home/smos/tools/l1pp*. Therefore L1PP_ROOT variable shall be set as */home/smos/tools/l1pp/* :

$ export L1PP_ROOT=/home/smos/tools/l1pp/

Remark: It is important to highlight that the environment variable $L1PP_ROOT shall be always terminated with a slash ("/") character. For instance if the L1PP is installed in */home/smos/tools/l1pp* the L1PP_ROOT shall be */home/smos/tools/l1pp/*.

It is also necessary to update the LD_LIBRARY_PATH environment variable with the location of the shared libraries, as follows:

$ export LD_LIBRARY_PATH=$L1PP_ROOT/lib64: $L1PP_ROOT/external_libs/lib64:$LD_LIBRARY_PATH

LD_LIBRARY_PATH shall include the directories where the different libraries are installed. Some of the external libraries are provided with the Installation Kit in the directory *L1PP_ROOT/external_libs/lib64,* while others shall be installed by the user. The path for the L1PP internal libraries (*L1PP_ROOT/lib64* – for the 64-bit version) shall also be included in LD_LIBRARY_PATH variable.

Note: If the user changes the location of the libraries, the LD_LIBRARY_PATH shall be updated accordingly. However, the files *dom4j.jar*, *jaxen-1.1-beta-8.jar* and *swing-layout-1.0.jar*, provided in *$L1PP_ROOT/lib* shall always be kept in this directory.

These two variables are updated if the prototype is executed with the scripts provided in the installation package. If the user wants to use a different execution method, these variables must be defined. In order to avoid overriding the user's local libraries, the L1PP will only use the provided additional libraries in case it cannot find them in the user-defined path.

3.5. Launching L1PP

The prototype may be started in 64bits by running the script *run-l1pp64.sh*, as follows:

$ sh ./run-l1pp64.sh

These scripts are provided as part of the installation package and are configured also during installation time. If the user moves the L1PP directory to another location, it shall be required to verify the correctness of the scripts.

The following type of output should be displayed right after execution:

```
Set L1PP_ROOT= /home/smostest/l1pp/
Set LD_LIBRARY_PATH=/home/smostest/l1pp//lib:/usr/local/lib/:/home/smostest/l1pp//external_libs/lib
Set XML_RW_API_HOME= /home/smostest/l1pp//dpgs_interface/
####################
Starting the L1PP GUI
####################
SMOS Level 1 Processor Prototype v3.4.0
Developed by Deimos Engenharia & Critical Software S.A.
Under contract of EADS CASA Espacio and ESA
```

In order to run in text mode the following command shall be executed:

$ sh ./run-l1pp64.sh -text

3.6. L1PP Configuration Files

The L1PP (3.4.0) has the ability to write and read products in the DPGS v3 format. It can be accomplished using two different interfaces: BinXML and XMLRW API (the default).

In the configuration file the key "**Science/Calibration products interface**" chooses the interface for reading/writing the products. The valid values are:

- 0: L1PP EEF format (no support for DPGS format)
- 1: DPGS V3 format using BinXML
- 2: DPGS V3 format using XML RW API (the default value)

Note: L1PP will process any products present in the L1A-IN directory irrespectively of the previous flag. Mixing DPGS products with EEF or XBAND products in L1A-IN may result in unforeseen results.

For reading the auxiliary data files (ADF) the configuration files also presents a choice. The key "**Auxiliary Data Files interface**" chooses the interface used for reading the ADFs. The valid values are:

- 0: L1PP EEF format (no support for DPGS format)

- 1: DPGS V3 format using BinXML
- 2: DPGS V3 format using XML RW API (the default value)

Note: The configuration files from release 1.6 and 1.6.1 are no longer compatible with the configuration files used in this release. An effort has been made to clean-up obsolete flags and propose final ones to be used during commissioning.

As in release 1.6.1, in the "config" directory for release 3.1.0, template configuration files are provided, since each ADF format needs a specific configuration file:

- configurationFileEEF.xml is prepared to handle the standard EEF L1PP ADFs
- configurationFileDPGS.xml is prepared to handle the V3 DPGS ADFs

In the default configuration, the configurationFile.xml is set to use DPGS ADFs. If the user wishes to use old/standard ADFs, the file configurationFileEEF.xml must be copied over configurationFile.xml. To revert this change, it suffices to copy configurationFileDPGS.xml over configurationFile.xml again (everytime this overwrite is done, the update_config_path script should be used to reset the user directories).

4. Compilation Procedure

The following sub-sections describe a set of procedures required to compile the L1 Processor Prototype and BinaryXML from the Release Kit.

4.1. Dependencies

In order to compile the BinaryXML library from the Release Kit, the following software must already be installed in the target platform:

- gcc 4.3 (http://gcc.gnu.org)
- Earth Explorer CFI 3.7.2 (http://eop-cfi.esa.int/)
- Libxml2 2.6.16 or higher
- BinX 1.2.6 (http://www.edikt.org/binx)
- xerces 2.8.0 (http://xml.apache.org/xerces2-j/) - required by BinX

In order to compile the SMOS L1PP, the following software must already be installed in the target platform:

- GCC 4.3 (http://gcc.gnu.org)
- glibc 2.3.4 (http://gcc.gnu.org)
- Earth Explorer CFI 3.7.2 (which already includes libxml2 2.6.16) (http://eop-cfi.esa.int/)
- BinaryXML File Handler 3.7 (http://www.smos.esa.int/)
- FFTW 3.1.2 (http://www.fftw.org/)
- BLAS v3.0 (http://www.netlib.org/blas/) (included in LAPACK distribution)
- LAPACK v3.0 (http://www.netlib.org/lapack/)
- DOM4J 1.5.2 (http://www.dom4j.org)
- Jaxen 1.1 (http://jaxen.codehaus.org)
- Log4c 1.2.0 (http://log4c.sourceforge.net)
- j2sdk1.5.0 (http://java.sun.com/j2se/1.4.2/download.html)
- XML RW API (ftp://193.146.123.163/smos/software/XML_RW_API/)

- xerces 2.7.0 (http://xml.apache.org/xercesc/) (needed by XML RW API)

Each product should be installed according to its own instructions in the target platform.

When all external libraries are installed, it is necessary to create some soft links to the installed libraries:

- create a link from the Explorer CFI home directory to "/opt/cfi"
- create a link from the BinX home directory to "/opt/binx"
- create a link from the xerces-c home directory to "/opt/xerces-c"
- create a link from the binxml-fh home directory to "/opt/binxml-fh"
- create a link from the XML RW API home directory to "/opt/xml_rw"

e.g.: An example for creating a link for the BinX library installed in the user's home directory would be:

$ ln -s ~/binx1.2 /opt/binx

If these links cannot be created, then the Makefile inside the l1pp directory should be updated with the paths to the different libraries by editing the first lines in the file. It should also be verified that the paths to the include and lib files are correct within the Makefile. It should also be verified that it has the permissions to access the libraries.

These instructions for compilation are valid for any OS (Linux, MacOS), as they are the paths where the Makefile expects to find the libraries.

For execution of the prototype, it is also necessary to update the LD_LIBRARY_PATH environment variable with the location of the shared libraries, as follows:

$ export LD_LIBRARY_PATH=/opt/xerces-c/lib:/opt/binxml-fh/lib:
/opt/cfi/aux_tools/libxml/LINUX/lib:$LD_LIBRARY_PATH

4.2. Compilation

The command “make clean all” shall initiate the compilation of all libraries and executables. Compilation on 32 or 64bits can be toggled by commenting out the variable ARCH64 inside the Makefile.

5. ANNEX A: L1PP INSTALL KIT CONTENTS

```
l1pp-3.4.0
├──bin
├──breakpoint
├──config
│   ├──iers_bulletins
│   ├──product_headers
│   └──xml_schemas
├──data
│   ├──adf-dpgs
│   ├──adf-eef
│   ├──l1a-in
│   ├──l1b-in
│   ├──l1c-in
│   ├──l1c-out
│   ├──processed-data
│   └──unprocessed-data
├──dpgs_interface
│   ├──lib32
│   │   └──MACIN
│   ├──lib64
│   │   └──LINUX
│   ├──projects
│   ├──smos
│   │   ├──config
│   │   ├──products
│   │   ├──schemas
│   │   └──tmp
│   └──xml_rw_api
│       └──headers
├──external_libs
│   ├──iono_models
│   │   ├──IGRF
│   │   └──IRI
│   ├──lib32
│   │   └──MACIN
│   └──lib64
│       └──LINUX
├──lib
├──lib64
│   └──LINUX
├──logs
└──scripts
```

6. ANNEX B: L1PP 3.4.0 USAGE OF DPGS ADF FILES

Product	Header	Datablock
APDL /APDS	Not used	Read
BFP		
BWGHT		
BSCAT		
DGG		
FAIL		
GALAXY		
GALNIR		
LCF		
LSMASK		
MASK		
MISP		
MOONT		
NIR		
PATT		
PLM		
PMS		
RFI		
SPAR		
SUNT		
VTEC		

7. ANNEX C: IMPLEMENTATION OF MANDATORY FIELDS IN L1C HEADERS

Item	L2 Usage	Value	Notes
Earth_Explorer_Header	L2 uses this value		Our headers are in line with the specification. The value defined by the schema is the one we use.
File_Name	L2 uses this value		This value is copied from the internal data-structure that holds the actual product name. The Site Instance value is always set to 0 in L1PP.
File_Description			This value is hard-coded as specified in the documentation.
Notes		x	No value is being written, but the schemas' default is '*x*'.
Mission	L2 uses this value	SMOS	It is always set to *'SMOS'*
File_Class	L2 uses this value		This value is copied from the product filename.
File_Type	L2 uses this value		This value is copied from the product filename.
Validity_Start	L2 uses this value		The value that is used to generate the product name is also used to calculate this value.
Validity_Stop	L2 uses this value		The value that is used to generate the product name is also used to calculate this value.

Item	L2 Usage	Value	Notes
File_Version	L2 uses this value		This value is copied from the product filename.
System		L1PP	Value obtained from L1PP configuration file.
Creator		L1PP	Always set to *'L1PP'*
Creator_Version			This value is set to the version of the L1PP. This same version is used to generate the product name.
Creation_Date			This value is generated when writing the header.
Ref_Doc		SO-TN-IDR-GS-0005	Always set to *'SO-TN-IDR-GS-0005'*
Acquisition_Station	L2 uses this value		L1PP propagates the values read from the previous levels. Currently, L0 products produced by SEPS-GS use the value *' VFR'*.
Processing_Centre		DMEP	Value obtained from L1PP configuration file.
Logical_Proc_Centre		DME	Value obtained from L1PP configuration file.
Orbit_Information	L2 uses this value		All *Orbit_Information* values are copied from previous levels products
Phase	L2 uses this value	+001	This value is read from previous level products (L0) and propagated. L0 products produced by SEPS-GS set this value to *'+001'*.

Item	L2 Usage	Value	Notes
Cycle	L2 uses this value	+000	This value is read from previous level products (L0) and propagated. L0 products produced by SEPS-GS set this value to *'+000'*.
Rel_Orbit	L2 uses this value	+00000	This value is read from previous level products (L0) and propagated. L0 products produced by SEPS-GS set this value to *'+001'*.
Abs_Orbit	L2 uses this value	+00000	This value is read from previous level products (L0) and propagated. L0 products produced by SEPS-GS set this value to *'+00000'*.
OSV_TAI/UTC/UT1	L2 uses this value	2007-06-04T 13:07:57.400000	This value is read from previous level products (L0) and propagated. L0 products produced by SEPS-GS set this value to *'2007-06-04T13:07:57.400000'*.
Leap_Second	L2 uses this value	2000-01-01T 00:00:00.000000	This value is read from previous level products (L0) and propagated. L0 products produced by SEPS-GS set this value to *'2000-01-01T00:00:00.000000'*.
X_/Y_/Z_Position	L2 uses this value	+0000000.000	This value is read from previous level products (L0) and propagated. L0 products produced by SEPS-GS set this value to *'+0000000.000'*.
X_/Y_/Z_Velocity	L2 uses this value	+0000.000000	This value is read from previous level products (L0) and propagated. L0 products produced by SEPS-GS set this value to *'+0000.000000'*.
Vector_Source	L2 uses this value	FP	This value is read from previous level products (L0) and propagated. L0 products produced by SEPS-GS set this value to *'FP'*.
Product_Confidence			This value is set according to the specification, however the error counters are not updated during the processing, so unless previous levels have a value different than 0, the Product_Confidence field will always be set to 'NOMINAL'.
SPH_Descriptor			Value hard-coded according to specification.

Item	L2 Usage	Value	Notes
Time_Info			
Precise_Validity_Start	L2 uses this value		It is computed for each header.
Precise_Validity_Stop	L2 uses this value		It is computed for each header.
Abs_Orbit_Start	L2 uses this value		It is computed for each header.
Start_Time_ANX_T	L2 uses this value		It is computed for each header.
Abs_Orbit_Stop	L2 uses this value		It is computed for each header.
Stop_Time_ANX_T	L2 uses this value		It is computed for each header.
UTC_at_ANX	L2 uses this value		It is computed for each header.
Long_at_ANX	L2 uses this value		It is computed for each header.
Ascending_Flag	L2 uses this value		This value is copied from previous levels. If this value is not available the value is set to 'A'
Semiorbit_Start_Time			This value is copied from previous levels. If this value is not available the value is set to 'UTC=2007-01-01T00:00:00.000000'
Semiorbit_Stop_Time			This value is copied from previous levels. If this value is not available the value is set to 'UTC=2007-01-01T00:00:00.000000'
Correlator_Layer			This value is copied from previous levels. If this value is not available the value is set to 'N'
Checksum			The value is correctly computed using the XMLRW methods.
Header_Schema	L2 uses this value		This value is correctly computed using the XMLRW methods.
Datablock_Schema	L2 uses this value		This value is correctly computed using the XMLRW methods.

Item	L2 Usage	Value	Notes
Header_Size			The value is correctly computed.
Datablock_Size			The value is correctly computed.
HW_Identifier		L1PP	The value is always set to 'L1PP'
Quality_Information			
Software_Error_Counter		0 (schema default)	Currently not available
Instrument_Error_Counter		0 (schema default)	Currently not available
ADF_Error_Counter		0 (schema default)	Currently not available
N_Discarded_Scenes		0	This value is not correctly computed, an internal **SPR** has been opened.
N_Invalid_Blocks		0 (schema default)	Currently not available
N_Missing_Packets		0 (schema default)	Currently not available
N_Missing_Blocks		0 (schema default)	Currently not available
Reference Datasets **(ALGORITHN_CONFIG_FILE)**			This value is always set to blanks in L1PP, since this file does not exist in L1PP.
Start_Lat	L2 uses this value		This value is correctly computed.
Stop_Lat	L2 uses this value		This value is correctly computed.
Start_Lon	L2 uses this value		This value is correctly computed.

Item	L2 Usage	Value	Notes
Stop_Lon	L2 uses this value		This value is correctly computed.
Mid_Lat	L2 uses this value		This value is correctly computed.
Mid_Lon	L2 uses this value		This value is correctly computed.
Radiometric_Accuracy_Scale	L2 uses this value	50	We use the reference value presented by the reference document. This value is taken from L1PP configuration file.
Pixel_Footprint_Scale	L2 uses this value	100	We use the reference value presented by the reference document. This value is taken from L1PP configuration file.
Percentage_of_Mixed_Pixels			This value is correctly computed.
Apodisation_Window		001	The value is not computed in L1PP, and it is always set to *'001'* - Blackman Window.
Total_Num_Grid_Points			This value is correctly computed.