SMOS L1OP 700

SMOS L1OP Processors

SOFTWARE USER MANUAL

Code : SO-SUM-DME-L1OP-0278
Issue : 1.5
Date : 09/07/2015

	Name	Function	Signature
Prepared by	SMOS IPF Team SMOS L1OP Team	Project Team	
Reviewed by	Antonio Gutierrez	Technical Consultant	
Approved by	Jose Barbosa	Project Manager	
Signatures and approvals on original			

DEIMOS Engenharia S.A.
Av. D. João II, Lote 1.17.01, Edifício Torre Zen, 10º
1998-023 Lisboa, PORTUGAL
Tel.: +351 21 893 3010 / Fax: +351 21 896 9099
E-mail: deimos@deimos.com.pt

This page intentionally left blank

Document Information

Contract Data	
Contract Number:	4000101241/10/I-AM
Contract Issuer:	IDEAS / ESA / ESRIN

Internal Distribution		
Name	Unit	Copies
IDEAS SMOS Team – DME	AS	1
Internal Confidentiality Level (DMS-COV-POL05)		
Unclassified ☐	Restricted ☑	Confidential ☐

External Distribution		
Name	Organisation	Copies
Manuel Martin-Neira	ESA	1
Steven Delwart	ESA	1
Raffaele Crapolicchio	ESA	1
Jon Swinton	IDEAS	1
Massimo Cardadci	IDEAS	1

Archiving	
Word Processor:	MS Word 2000
File Name:	SO-SUM-DME-L1OP-0278-v700-Software-Users-Manual_E1_R5.doc

Document Status Log

Issue	Section	Change description	Date	Approved
1.0		First issue of the document, for v600.	02/05/13	
1.1		Updated for v602, added clarification in Section 4.2.2.2 on schema update procedure. Corrected typo in schema package version.	05/08/13	
1.2		Added clarification on $NRTP_CONFIG setting in Section 4.2.2	11/09/13	
1.3		Updated L1OP schemas version reference	25/03/14	
1.4		Updated memory requirements and schema reference to one produced during FAT	09/04/14	
1.5		Updated for L1OP v700	09/07/15	

Table of Contents

List of Tables

1. INTRODUCTION

This project concerns the design and development of the several PFW Processors derived from L1PP and NRTP Code (L1a, L1b, L1c, FTT).

The final objective is the integration of algorithms from the L1 Prototype Processor into the DPGS.

1.1. Purpose

This document describes the procedures for installing the SMOS PFW Processors derived from the L1PP and NRTP (L1b, FTT) executables.

The document also describes the orchestration principles, configuration and error / warning messages.

1.2. Document Structure

Section 1 gives a short introduction to the system and to this document

Section 2 lists the applicable and reference documents.

Section 3 describes the overall context.

Section 4 focuses on PFW Processors, including the Installation Procedure, the Orchestration and Errors and Warnings.

1.3. Intended Readership

This installation manual is intended to be read and used by the personnel who are responsible for installing the system on-site, as well as by the system maintenance and operations personnel. Minimum Linux system knowledge is expected from the operator in order to understand and follow the procedures depicted in this document.

1.4. Acronyms and Abbreviations

The acronyms and abbreviations used in this document are the following ones:

Acronym	Description
AD	Aplicable Document
ADF	Auxiliary Data File
API	Application Programming Interface
CFI	Customer Furnished Item
COTS	Commercial Off-The-Shelf
DMS	DEIMOS Space
DPGS	Data Processing Ground Segment
ESA	European Space Agency
GSL	General Software Library
HW	Hardware
I/F	Interface
ICD	Interface Control Document
L1OP	SMOS Level 1 Operational Processor
L1PP	SMOS Level 1 Prototype Processor
NIR	Noise Injection Radiometers
NRT	Near Real Time
NRTP	Near Real Time Processor
O/S	Operating System
PDPC	Payload Data Processing Centre
PDR	Preliminary Design Review
PFW	Processing FrameWork
RD	Reference Document
RPF	Reference Processing Facility
SUM	Software User Manual
SMOS	Soil Moisture and Ocean Salinity
SW	Software
TBC	To Be Confirmed
TBD	To Be Defined / Decided

2. RELATED DOCUMENTS

2.1. Applicable Documents

The following table specifies the applicable documents that shall be complied with during project development.

Table 1: Applicable documents

Reference	Code	Title	Issue
[SOW]	SO-SOW-ESA-GS-6647	SMOS Expert Support Laboratories for the period 2010-2014 - ESL Level 1 Calibration and Reconstruction	**1.2** *10/12/2009*
[A-1]	SO-TN-IDR-GS-0003	SMOS Level 0 Product specification	3.5
[A-2]	PE-TN-ESA-GS-0001	Earth Explorer File Format Standard	1.4
[A-3]	XSMS-GSEG-EOPG-TN-05-0006	Tailoring of the Earth Explorer File Format Standard for the SMOS Ground Segment	1.0
[A-4]	IDEAS+-SER-MGT-SPE-0792	IDEAS+ SMOS IDEAS ICD	***1.11*** *20/02/2015*
[A-5]	SO-TN-IDR-GS-0005	SMOS Level 1 and Auxiliary Data Products Specification	**5.35** *(preliminary release 30/06/14)*
[A-6]	SO-TN-IDR-GS-0006	SMOS Level 2 and Auxiliary Data Products Specification	7 *(14/12/2012)*
[A-7]	SO-MA-IDR-GS-0004	SMOS DPGS XML Schema Guidelines	2.1
[A-8]	SO-TN-IDR-GS-0011	SMOS DPGS Reports Specification	2.5
[A-9]	ECSS-E-40 Part 1B	Space Engineering – Software – Part 1: Principles and Requirements	-
[A-10]	ESA-ID-ACS-GS-0001	PDS-IPF ICD Generic Interface Guidelines	1.1
[A-11]	SO-DS-DME-L1OP-0007	SMOS L0 to L1a Detailed Processing Model and ATBD	**2.19** *07/09/15*
[A-12]	SO-DS-DME-L1OP-0008	SMOS L1a to L1b Detailed Processing	**2.20**

Reference	Code	Title	Issue
		Model and ATBD	*07/09/15*
[A-13]	SO-DS-DME-L1OP-0009	SMOS L1b to L1c Detailed Processing Model and ATBD	***2.14*** *25/03/14*
[A-14]	ECSS-E-40 Part 2B	Space Engineering – Software – Part 2: Document Requirements Definition	-
[A-21]	XSMS-GSEG-EOPG-TN-09-0004	SMOS L1OP-V3 Orchestration Baseline	2.8 *(20/06/13)*

2.2. Reference Documents

The following table specifies the reference documents that shall be taken into account during project development.

Table 2: Reference documents

Reference	Code	Title	Issue
[R-1]	EEOM-SMOS-MRD	Mission Objectives and Scientific Requirements of the Soil Moisture and Ocean Salinity	5.0
[R-2]	SO-DS-IDR-GS-0001	DPGS System Technical Description & Operations Concept	2.5
[R-3]	SO-RS-ESA-SYS-0555	SMOS System Requirements Document	4.2
[R-4]	SO-TN-IDR-GS-0010	SMOS DPGS Acronyms	1.11
[R-5]	SO-ID-IDR-GS-0009	XML R/W API Software User Manual	2.1
[R-7]	SO-TN-IDR-GS-0013	SMOS PDPC Core Level 1 Processor Orchestration Technical Note	2.3
[R-10]	SO-MA-IDR-GS-1002	SMOS DPGS General software Library User Manual	1.9 *26/06/09*
[R-11]	CS-MA-DMS-GS-0002	Earth Explorer CFI Software: General SUM	3.7
[R-12]	CS-MA-DMS-GS-0003	Earth Explorer CFI Software: EXPLORER_LIB Software User Manual	3.7

Reference	Code	Title	Issue
[R-13]	CS-MA-DMS-GS-0004	Earth Explorer CFI Software: EXPLORER_ORBIT Software User Manual	3.7
[R-14]	CS-MA-DMS-GS-0005	Earth Explorer CFI Software: EXPLORER_POINTING Software User Manual	3.7
[R-15]	CS-MA-DMS-GS-0006	Earth Explorer CFI Software: EXPLORER_VISIBILITY Software User Manual	3.7
[R-16]	EE-MA-DMS-GS-0007	Earth Explorer CFI Software: EXPLORER_DATA_HANDLING SUM	3.7
[R-17]	CS-MA-DMS-GS-0008	Earth Explorer CFI Software: EXPLORER_FILE_HANDLING SUM	3.7
[R-27]	SO-ID-IDR-GS-1007	PDPC core PFW ICD	1.4
[R-29]	MINARC-DMS-TECADD01-R	MINARC Mini-Archive Architecture and Design document.	1.0

3. OVERALL CONTEXT

The following elements of the L1 processor are to be considered:

- L1a processors, to be deployed under the PDPC-Core/PFW
- L1b processor for Dual Polarization data, to be deployed under the PDPC-Core/PFW
- L1b processor for Full Polarization data, to be deployed under the PDPC-Core/PFW
- L1c processor, to be deployed under the PDPC-Core/PFW
- Flat Target Transform, to be deployed under the PDPC-Core/PFW

4. PFW PROCESSORS

4.1. Package Content

The full package consists of one single self-extractable packet:

- SMOS_L1OP_Proc_VV_VV__<centre>_yyyymmddhhmm.sh

Where:

VV_VV: version of the L1OP delivery.

<centre>: Target of the package. FPC, RPF or LTA.

yyyymmddhhmm: Package creation date, year, month, day, hour, minute.

The package contains all the PFW processors that are included as part of the delivery. Nominally the processors included will be:

- HKTM1A_VV_VV
- CAL_1A_VV_VV
- NIRCAL_VV_VV
- SCIL1A_VV_VV
- PRL1BD_VV_VV
- PRL1BF_VV_VV
- FTTGEN_VV_VV
- SCI_1C_VV_VV

Task tables are also included, as part of the package, upon installation the **user will be requested** if the Task Tables shall be extracted and in case of affirmative answer, the user will be requested to include the path that shall be used. These set of task tables are unique for each delivery and the operator must place them in the correct destination facility.

Besides, an un-installation script is also provided with each delivery.

4.2. Installation Procedure

Two versions of the installation package are delivered:

- PFW
- RPF (no differences with respect to PFW).
- LTA (Currently no difference, the only foreseen difference is the possibility of including different Task Tables with respect to the PFW)

4.2.1. Versions

- The PFW Processor executables are built using several modules, including CFIs. They share many of the NRTP libraries and their dependencies. Each executable is compiled in a 64 bit machine under gcc 4.1.2 and all DEIMOS libraries are linked statically. The Earth Explorer Mission Software CFI is also linked statically while dynamic libraries are used for all remaining third party COTS.

Table below list the relevant COTS and CFI used when compiling/executing the executables, together with their version:

Name	Static [S] / Dynamic [D]	Delivered	Version	Source
Red Hat	-	NO	Linux ES 5.0	
GNU C++ Compiler	-	NO	4.1.2	http://gcc.gnu.org
GNU libc library	Never delivered	NO	2.5.18	http://gcc.gnu.org
Earth Explorer CFI Software	[S]	NO	3.7.3	http://eop-cfi.esa.int/CFI
XML R/W API (libxrwa)	[D]	YES	4.2.1	ftp://131.176.251.166/smos/software/XML_RW_API
SMOS schemas	-	NO	06-02-03	ftp://131.176.251.166/smos/schemas
GSL (libprocgsl)	[D]	YES	1.8	ftp://131.176.251.166/smos/software/GSL
libxerces-c	[D] [D]	YES	2.7.0 (used by XML R/W API) 2.8.0 (used by VEGA RW)	ftp://131.176.251.166/smos/software/xerces http://xerces.apache.org/xerces-c/download.cgi
libOpenThreads	[D]	YES	1.2	http://openthreads.sourceforge.net/
libgfortran	[D]	YES	4.2.2	http://gcc.gnu.org
libgomp	[D]	YES	3.0	http://openmp.org

Table 3: COTS and CFIs

The LD_LIBRARY_PATH environment variable has to be updated in order to include the location of the installed dynamic libraries. The path is added at the end of the environment library (see section 4.2.2.1 for details on the use of environment variables). In the previous table the column Delivered indicates whether the COTS is provided or not, in case it is not it should be already present in the system.(eg:R/W API).

4.2.2. Installer execution

The installer file delivered as part of the release is to be copied to the target machine in order to start the installation process.

The installer is provided in Shell script form and it can be executed in a standard way; from the same folder type:

$> ./SMOS_L1OP_Proc_VV_VV__centre_yyyymmddhhmm.sh

Before the execution, however, the environment variable $NRTP_CONFIG must be defined. The user can simply execute the script "setL1OP.sh", which automatically sets all needed environment variables. Once executed, the installer will ask for a number of directories in which the binaries and libraries files will be copied. Also, the installer will ask whether the operator wants the installer to copy configuration files (iono models and rw files) to directory pointed by the variable $NRTP_CONFIG (default is not to copy).

The installer offers default paths but it is also possible to specify them manually. The default path for the processors is:

`/application/smos/install/processors/<processor_name>/<processor_version>`
`/<executables_file_names>`

Libraries would be installed in:
`/application/smos/install/processors/L1OPlibraries/<processor_version>`
`/<librariesfiles>`

CNF_xxxxxx file (a different filetype is used per processor) would be installed in:
`/application/smos/install/processors/<processor_name>/<processor_version>`
`/config/<CNF_xxxxxx file`

Additionally, the installer generates a profile file with the required environment variables, see section 4.2.2.1, placing it in the processors path. Note that this profile is NOT loaded during the installation and that the user has to verify that the variables listed in them are loaded before running the processor.
It also copies a number of additional configuration files under `/application/smos/install/processors/<processor_name>/<processor_version>` (see section 4.2.2.2)

Finally, the installer process generates a special file used by the uninstaller script containing the directories and files created during the installation process. This file is stored into the folder:

~/.l1op_smos_installations/VV_VV.dat, where VV_VV is the installation version.

During the uninstall process, if there were more than one version installed, the un-installation process would display them and the operator will have to choose the version to uninstall.

Final checks:

The CEC processors and the PFW processors are based on a two-executables architecture: <executable_file_name> and <CHILD_executable_file_name>, both of them placed under `/application/smos/install/processors/<processor_name>/<processor_version>`.

Parent processor has two ways to know where its child is:

- First, it checks if the environment variable CHILD_<processor_name>_<processor_version> exists (see section 4.2.2.1). This variable is used to contain the path in which the executables are placed.
- If this environment variable does not exist, then parent processor looks for its child in the current directory.

In releases prior to 03_31, the PFW did not require a dedicated environment variable. The same behaviour can be left after 03_31, just by not defining the CHILD_<processor_name>_<processor_version> environment variable.

4.2.2.1. Environment Variables

The installer will have created a dedicated profile file (`profile_smos_vv_vv`) with the definition of the environment variables to be used by the executable. We make no assumptions on the existence of other Software items in the same system/user. This profile should be added to the pre-existing profiles as needed. The required environment variables are:

- LD_LIBRARY_PATH: in order to ensure the correct selection of dynamically linked libraries, the path for the executable libraries shall be included in the environment variable. L1OP needed values are the ones that can be found in the delivered profile.
- NRTP_CONFIG: is used to point to the **VEGA read-write library configuration file** (see next section). The variable must be defined such that it point to the location of `/rw/` vega_rw_api.usr_conf.xml (nominally NRTP_CONFIG would be `/application/smos/install/processors/<processor_name>/<processor_version>`) The variable does not include the `/rw/` folder nor the VEGA filename. This variable must also point to the root path of the ionosphere models that will be used in the processing. They are included in the delivery and deployed with the correct folder name (**iono_models**).
- CHILD_<processor_name>_<processor_version> (ex: CHILD_PRL1BD_03_31): This environment variable is specific of each delivery. It is used to point to the path in which the executables are placed and its value is assigned dynamically during the Installer execution.

  This environment variable is only needed by the RPF in order to let the parent locate the child executable locations. It also allows to have in the same machine installed the FWFGMA and the L1b processors.

  By default, this environment variable is commented. In the cases in which this is needed, for example RPF, the user must edit the profile and update the line where this variable is set.

4.2.2.2. Configuration Files

The installer copies a number of configuration files into dedicated folders:

- `xml_rw_api.usr_conf.xml`, in the executable folder. This file is only needed in case the executable is launched from the command line (see section 4.2.2.4 for more details).
- `vega_rw_api.usr_conf.xml`, in a folder named `rw` within the configuration folder. This file is needed by the VEGA read-write library. Shall be located at $NRTP_CONFIG/rw
- `CNF_xxxxxx` is copied in the executables folder under a folder named "config" as specified in section 4.2.2, and is provided **as a reference** in order to update the operational `CNF_xxxxxx` with the processor related parameters (processor version). As an example the configuration reference provided for the PRL1BD will be located at (depending on the install path used) `/application/smos/install/processors/PRL1BD/vv_vv/config/`

  Together with each delivery a set of task tables are also provided. Each of them contains, as an example, the path in which should be placed the CNF_xxxxxx corresponding to each processor. The operator will have to take charge of configure correctly this route.

  The following section shows an example of this configuration file which is provided with each delivery within its list of configurable parameters.

The first configuration file shall actually be copied to the folder from which the executables are invoked by the PFW.

The system uses the VEGA read-write library inherited from the NRTP. This library allows DEIMOS to adapt Indra read-write library and to refill the structures needed by the processors in order to process the data and get correct products.

The vega_rw_api.usr_conf.xml file describes the version of Indra's schemas that it uses. Nominally it uses the last schema version of every product (LV).

> Care must be taken when new schemas are installed. The vega_rw_api.usr_conf.xml has to be updated to use the correct version of the schemas whenever the new schemas have a higher version number than the one included in the L1OP release.

Nevertheless, the system crosschecks that the version used by the VEGA library is aligned to the one in the CNF_xxxxxx issuing a warning in case it is not the case.

4.2.2.3. CNF_xxxxxx Configuration File

4.2.2.3.1.1. Configurable Parameters

Some fields in CNF_xxxxxx must be configured by the user. The list of configurable parameters is included below. For the complete description of this file, see specification [AD.5]

Tag Name	Description	Comments
Processing_Centre	ID code of the Processing Centre that has generated the product {ESAC, others TBD –e.g. LTA location-}. This is the physical location where the product is generated. L2Ps do not obtain this tag from same tag in L1C products.	Is needed to select the processing center in which L1OP is running
Logical_Proc_Centre	ID code of the Logical Processing Centre that has generated the product. The Logical Processing Centre is the group of subsystems within the Processing Centre working co-ordinately to generate the product. Possible values are: {FPC}: SMOS DPGS Fast Processing Centre @ ESAC; {LTA}: SMOS DPGS LTA @ Kiruna; {CEC}: SMOS DPGS Calibration & Expertise Centre @ESAC; {IDR}: Indra ; {GMV}: GMV; {INS}: INSA L2Ps do not obtain this tag from same tag in L1C products.	Is needed to select the appropiate value from [AD.04]
Verbose_Mode	Flag to run the processors in verbose mode or not. Verbose mode shows more trace log information (thread, file, method and line). (0: OFF/1: ON)	

Tag Name	Description	Comments
Conversion_Files_Directory	Directory where conversion files are stored. These files are needed in Gmatrix generator to convert product formats (SMOS format to Prototype format and vice versa)	Never used, Prototype format is dicontinued
List_of_Hosts	Tag starting a list of hosts which contain L1OP installations, specifying the number of threads for each Processor in each host.	
Host_Name	Host name for the Processors are installed, as identified by PDPC-Core. It is a logical name with DNS, not an IP direction	Unless the Host_Name will be correctly configured, Some fields of the output product header would be empty.
HW_Identifier	Unique identifier of the hardware involved in the processing.“nnnn” where n are digits or characters.L2Ps do not use this tag from L1C SPH.	
Number_of_Threads	Number of threads that the Processor must start in each execution	For Example: 4 threads for a dual core CPU host.
TEC_File_Path	Path to the directory where IRI model files are stored	Not used from versions 351 onwards. Use env variable NRTP_CONFIG instead. See section 4.2.2.5
IGRF_File	Complete path and filename for IGRF file	Not used from versions 351 onwards. Use env variable NRTP_CONFIG instead. See section 4.2.2.5

Table4: List of CNF_xxxxxx configurable parameters

Note: The field “Host_Name” must be configured with the same name that you obtain when you execute the following command in the server in which processors are installed.

```
$> hostname –s
```

If this field is not configured correctly, then, the output product header will contain some empty values, such <HW_Identifier></HW_Identifier>.

4.2.2.3.1.2. Example of CNF_xxxxxx file

<Earth_Explorer_File>

```
<Earth_Explorer_Header xmlns="http://193.146.123.163/smos/schemas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://193.146.123.163/smos/schemas
HDR_SM_XXXX_CNF_FTTGEN_0200.xsd">
  <Fixed_Header>

  <File_Name>SM_TEST_CNF_PRL1BD_20050101T000000_20500101T000000_303_002_1.EEF</
File_Name>
    <File_Description>Internal configuration file for SMOSL1 Operational
Processor</File_Description>
  <Notes></Notes>
  <Mission>SMOS</Mission>
  <File_Class>TEST</File_Class>
  <File_Type>CNF_PRL1BD</File_Type>
  <Validity_Period>
   <Validity_Start>UTC=2005-01-01T00:00:00</Validity_Start>
    <Validity_Stop>UTC=2050-01-01T00:00:00</Validity_Stop>
  </Validity_Period>
  <File_Version>0002</File_Version>
  <Source>
   <System>DPGS</System>
    <Creator>L1OP</Creator>
    <Creator_Version>301</Creator_Version>
    <Creation_Date>UTC=2008-09-24T17:00:00</Creation_Date>
  </Source>
 </Fixed_Header>
 <Variable_Header>
  <Specific_Product_Header>
   <Ref_Doc>SO-ID-IDR-GS-0008</Ref_Doc>
   <Total_Size>00000057665</Total_Size>
  </Specific_Product_Header>
 </Variable_Header>
</Earth_Explorer_Header>
<Data_Block type="xml" xmlns="http://193.146.123.163/smos/schemas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xsi:schemaLocation="http://193.146.123.163/smos/schemas
DBL_SM_XXXX_CNF_FTTGEN_0200.xsd">
  <Environment_Parameters>
    <Processing_Centre>IDMP</Processing_Centre>
      <Logical_Processing_Centre>1</Logical_Processing_Centre>
        <Byte_Order>0123</Byte_Order>
        <Verbose_Mode>1</Verbose_Mode>
        <Conversion_Files_Directory>/application_cache/</Conversion_Files_Directory>
        <System>DPGS</System>
        <Creator_Version>303</Creator_Version>
  </Environment_Parameters>
  <Engineering_Parameters>
    <List_of_Hosts count="1">
      <Host>
        <Host_Name>DPGSCorePfw-1</Host_Name>
        <HW_Identifier>0001</HW_Identifier>
        <List_of_Multithreaded_Processors count="1">
          <Multithreaded_Processor>
            <Processor_Name>PRL1BD</Processor_Name>
            <Processor_Version>03_03</Processor_Version>
            <Number_of_Threads>004</Number_of_Threads>
          </Multithreaded_Processor>
        </List_of_Multithreaded_Processors>
      </Host>
    </List_of_Hosts >
    <List_of_Supported_Schemas count="99">
      <Schema>
        <Product_Type>SM_XXXX_TLM_MIRA1A</Product_Type>
        <Schema_Version>0201</Schema_Version>
      </Schema>
    </List_of_Supported_Schemas>
    <TEC_File_Path>/application/smos/install/processors/SCI_1C/03_00/tec_model/</TEC_File_Path>
```

<IGRF_File>/application/smos/install/processors/SCI_1C/03_00/geomag_model/igrf10.dat</IGRF_File>

</Engineering_Parameters>

</Data_Block>

</Earth_Explorer_File>

4.2.2.4. Confidence Tests

It is possible to execute a number of tests to ensure the correct installation of the executable. These confidence tests are performed by executing the processor from the command line, directly within the folder where the executable is placed.

The executable needs to have in the directory from which it is executed the R/W configuration file (`xml_rw_api.usr_conf.xml` already created by the installer). This file has to be modified such that internally it points to the location of the XML Schemas.

The environment variable XML_RW_API_HOME has to exist and point to XMLRWAPI.

Once the file is modified it is possible to run test cases, by executing the following commands:

./$EXECUTABLE_PATH/<*processor_Name*>_0x_yy.exe $TDS_PATH/job_order_name > log.txt

where $TDS_PATH should be the location of the processor test cases (full path up to the job order).

For the tester convenience a number of TDS will be delivered together with each release of the processors. This TDS shall be agreed in advance to the delivery in order to cover the specifics of each delivery. DME will provide the job orders and (when available) a number of scripts to run different scenarios one after the other. The TDS will be hard-coded to a specific location that will be indicated with the TDS release. The path of the location can be easily simulated using soft links. In case the same location can not be used at the host machine, the provided job orders and scripts will need to be manually modified by the operator in charge of the running the test.

4.2.2.5. Changing the iono-models used by the processor.

From release v351 onwards, the iono-models **are delivered together with the processors**. As explained before the Configuration parameters “TEC_File_Path” and “IGRF_File” in the configuration file are no more used by the processor. Instead the environmental variable “NRTP_CONFIG” will be used, under that variable a folder named “iono_models” must exist containing the different models in the following way:

$NRTP_CONFIG/iono_models/
IGRF/
IGRF.unx
IRI/
<iri_files>

This structure is created in the appropriate folder by the installer, however it is simple to modify the iono-models used in case of need. The recommended procedure would be as follows. With no processors running:

$> cd $NRTP_CONFIG

$> mv iono_models iono_models_vv_vv (where vv_vv could be the release version)

$> mkdir iono_models_XX_XX (where XX_XX coulde be the new version id for the models)

copy inside iono_models_XX_XX the IGRF and IRI files in the same folder structure described above.

$> ln –s iono_models_XX_XX iono_models

To switch between different versions the link can be easily changed.

4.3. Orchestration

For this section, please refer to [A.21] and [R7].

4.4. Reference Data Sets and Product Usage

Depending on the value of a number of flags in the AUX_CNFL1P some of the input files might not be used by the algorithm. In this case the Reference Data Sets will not list them either.

4.4.1. L1b processor for Dual/Full Polarization

Within **MIR_SC_x1B**:

MDS are:

- TEMP_SNAPSHOT_FULL (for full products)
- TEMP_SNAPSHOT_DUAL (for dual products)
- SCENE_BIAS_CORRECTION

RDS are:

- TLM_MIRA1A
- MIR_SC_x1A
- MIR_GMATx not used if;

  flat_correction_type equal 1 and
  direct_sun_correction_type equal 0 and
  reflected_sun_correction_type equal 0 and
  direct_moon_correction_type equal 0 and
  earth_contribution_removed equal 0 and
  sky_contribution_removed equal 0

- MIR_JMATx not used if:

  reconstruction_image_algorithm equal 0

- MIR_FTTx __ not used if:

  flat_correction_type equal 0

- AUX_PATT__ not used if:

  reconstruction_image_algorithm > 0 and
  backlobe_contribution_removed equal 0

- AUX_GALAXY not used if
  sky_contribution_removed equal 0
- AUX_PLM___
- AUX_SUNT__ not used if:
  reflected_sun_correction_type equal 0 and
  direct_sun_correction_type equal 0
- AUX_MOONT_ not used if:
  direct_moon_correction_type equal 0
- AUX_BFP___
- AUX_BWGHT_
- AUX_BSCAT_ not used if:
  reflected_sun_correction_type equal 0
- AUX_CNFL1P
- MPL_ORBSCT
- AUX_DGG___ not used if:
  earth_contribution_removed equal 0 and
  reflected_sun_correction_type equal 0
- AUX_LSMASK not used if:
  earth_contribution_removed equal 0 and
  reflected_sun_correction_type equal 0
- AUX_FAIL__

For **MIR_TARx1B**:

MDS are:

- TEMP_SNAPSHOT_FULL (for full products)
- TEMP_SNAPSHOT_DUAL (for dual products)
- SCENE_BIAS_CORRECTION

RDS are:

- TLM_MIRA1A
- MIR_TARx1A
- MIR_GMATx not used if;
  flat_correction_type equal 1 and

direct_sun_correction_type equal 0 and

reflected_sun_correction_type equal 0 and

direct_moon_correction_type equal 0 and

earth_contribution_removed equal 0 and

sky_contribution_removed equal 0

- MIR_JMATx not used if:
  reconstruction_image_algorithm equal 0
- MIR_FTTx __ not used if:
  flat_correction_type equal 0
- AUX_PATT__ not used if:
  reconstruction_image_algorithm > 0 and
  backlobe_contribution_removed equal 0
- AUX_GALAXY not used if
  sky_contribution_removed equal 0
- AUX_PLM___
- AUX_SUNT__ not used if:
  reflected_sun_correction_type equal 0 and
  direct_sun_correction_type equal 0
- AUX_MOONT_ not used if:
  direct_moon_correction_type equal 0
- AUX_BFP___
- AUX_BWGHT_
- AUX_BSCAT_ not used if:
  reflected_sun_correction_type equal 0
- AUX_CNFL1P
- MPL_ORBSCT
- AUX_DGG___ not used if:
  earth_contribution_removed equal 0 and
  reflected_sun_correction_type equal 0
- AUX_LSMASK not used if:
  earth_contribution_removed equal 0 and

reflected_sun_correction_type equal 0

- AUX_FAIL__

4.4.2. Flat Target Transform

Within MIR_FTTx__:

MDS is

- Flat_Target_Transformation

RDS are:

- TLM_MIRA1A
- MIR_TARx1A
- AUX_GALNIR
- AUX_BWGHT_
- AUX_BFP___
- AUX_CNFL1P
- MPL_ORBSCT
- AUX_MISP__
- MIR_FTTx__

4.5. Errors and Warnings

The execution of the processor can result in the following errors and warnings:

Error Description	Error Code	Log Message Shape	Variable Content	Processor Reaction			
				Program Behavior	Trace	Exception	Exit program
CNF_L1OP__ not available as input	MISSING_CONFIGURATION_FILE	YYYY-MM-DD hh:mm:ss.nnn XXXXXX_02_00[ppppppppppp]: [E] (MISSING_CONFIGURATION_FILE) CNF_L1OP__ (%s) is not available as input. %s : It is the CNF_L1OP file name.	CNF_L1OP file name.	Error log message generated. Abort L1OP.	error	runtime_error	Yes
Any of the mandatory input products is not listed in the job order or is not physically available	MISSING_MANDATORY_INPUT	YYYY-MM-DD hh:mm:ss.nnn XXXXXX_02_00[ppppppppppp]: [E] (MISSING_MANDATORY_INPUT) Mandatory file (%s1) is missing. %s2 %s1 : It is the file type missing. %s2 : Explain the cause: 1) It is not listed in the job order. 2) It is not physically available.	Missing File Type. Cause.	Error log message generated. Abort L1OP.	error	failure	Yes
Any error from XML RW API that may lead the L1OP to abort (e.g. schema not found, error when closing product)	FATAL_ERROR_FROM_XML_RW_API	YYYY-MM-DD hh:mm:ss.nnn XXXXXX_02_00[ppppppppppp]: [E] (FATAL_ERROR_FROM_XML_RW_API) %s %s : Error message returned back from XML RW API library.	Error message from XML RW API Library	Error log message generated. Abort L1OP.	error	failure	Yes
Generic error that does not fit any of the previous codes	UNDEFINED_ERROR	YYYY-MM-DD hh:mm:ss.nnn XXXXXX_02_00[ppppppppppp]: [E] (UNDEFINED_ERROR) %s %s : Error message.	Error Message as returned by NIRACL algorithm	Error log message generated. Abort L1OP.	error	Depends on error.	Yes
One product can not be generated. It can be any type of product, even a browse product.	NO_PRODUCT_GENERATED	YYYY-MM-DD hh:mm:ss.nnn XXXXXX_02_00[ppppppppppp]: [W] (NO_PRODUCT_GENERATED) The product (%s1) can not be generated due to: %s2 %s1 : Product not generated. %s2 : Reason not to generate it.	Product not generated. Reason not to generate it.	Skip product generation and continue. Send warning message.	warning	No	No

	Error Description	Error Code	Log Message Shape	Variable Content	Processor Reaction			
					Program Behavior	Trace	Exception	Exit program
5	Any warning that does not fit any of the previous codes	UNDEFINED_WARNING	YYYY-MM-DD hh:mm:ss.nnn XXXXXX_02_00[ppppppppppp]: [W] (UNDEFINED_WARNING) %s %s : Warning message.	Warning message as returned by NIRACL algorithm	Send warning message. Continue processing.	warning	No	No
7	Memory needed to process not available to L1OP	NRT_ERR_CODE_ERROR ALLOCATING_MEMORY	YYYY-MM-DD hh:mm:ss.nnn XXXXXX_02_00[ppppppppppp]: [E] (UNDEFINED_ERROR): Error allocating memory.	Error allocating memory.	Error log message generated. Abort L1OP.	error	No	Yes
8	Error reading a file or a field from a file.	NRT_ERR_CODE_ERROR READING_FILE	YYYY-MM-DD hh:mm:ss.nnn XXXXXX_02_00[ppppppppppp]: [E](FATAL_ERROR_FROM_XML_RW_API) Error reading file %s	Error Message as returned by RW API	Error log message generated. Abort L1OP.	Eror	No	Yes
9	Input value is out of bounds	NRT_ERR_CODE_VALUE OUT_OF_BOUNDS	YYYY-MM-DD hh:mm:ss.nnn XXXXXX_02_00[ppppppppppp]: [W] (UNDEFINED_WARNING) %s %s : Warning message.	Warning Message as returned by algorithms API	Warning log message generated.	Warning	No	No
10	Input data is incomplete	NRT_ERR_CODE INCOMPLETE_DATA	YYYY-MM-DD hh:mm:ss.nnn XXXXXX_02_00[ppppppppppp]: [E] (UNDEFINED_ERROR) %s %s : Error message.	Error Message as returned by algorithms API	Error log message generated. Abort L1OP.	Error	No	Yes
11	IGRF Model not found or incorrect	NRT_ERR_CODE_ERROR IGRF_MODEL	YYYY-MM-DD hh:mm:ss.nnn XXXXXX_02_00[ppppppppppp]: [E] (UNDEFINED_ERROR) %s %s : Error message.	Error Message as returned by algorithms API	Error log message generated. Abort L1OP.	Error	No	Yes
12	IRI Model not found or incorrect	NRT_ERR_CODE_ERROR IRI_MODEL	YYYY-MM-DD hh:mm:ss.nnn XXXXXX_02_00[ppppppppppp]: [E] (UNDEFINED_ERROR) %s %s : Error message.	Error Message as returned by algorithms API	Error log message generated.	Error	No	No
13	Input data used is not valid.	NRT_ERR_CODE_ERROR DATA_CORRUPTED	YYYY-MM-DD hh:mm:ss.nnn XXXXXX_02_00[ppppppppppp]: [E] (UNDEFINED_ERROR) %s	Error Message as returned by algorithms API	Error log message generated. Abort	Error	No	Yes

	Error Description	Error Code	Log Message Shape	Variable Content	Processor Reaction			
					Program Behavior	Trace	Exception	Exit program
			%s : Error message.		L1OP.			
4	Mathematical exception resulting from a division by zero.	NRT_ERR_CODE_ERROR DIVISION_BY_ZERO	YYYY-MM-DD hh:mm:ss.nnn XXXXXX_02_00[ppppppppppp]: [E] (UNDEFINED_ERROR) %s %s : Error message.	Error Message as returned by algorithms API	Error log message generated. Abort L1OP.	Error	No	Yes
5	Number of iterations exceeded maximum value in Newton Raphson method to compute Raw Quadrature measurement.	NRT_ERR_CODE_ERROR NO_CONVERGENCE	YYYY-MM-DD hh:mm:ss.nnn XXXXXX_02_00[ppppppppppp]: [E] (UNDEFINED_ERROR) %s %s : Error message.	Error Message as returned by algorithms API	Error log message generated. Abort L1OP.	Error	No	Yes
6	Any error or warning resulting from EE CFI usage that does not abort the processing.	NRT_ERR_CODE ERROR_FROM_CFI_LIB	YYYY-MM-DD hh:mm:ss.nnn XXXXXX_02_00[ppppppppppp]: [W] (UNDEFINED_WARNING) %s %s : Warning message.	Warning Message as returned by EE API	Warning log message generated.	Warning	No	No
7	Data conversion problem	NRT_ERR_CODE_ERROR CONVERTING_DATA	YYYY-MM-DD hh:mm:ss.nnn XXXXXX_02_00[ppppppppppp]: [E] (UNDEFINED_ERROR) %s %s : Error message.	Error Message as returned by algorithms API	Error log message generated. Abort L1OP.	Error	No	Yes
8	Problem accessing binxml library	NRT_ERR_CODE_ERROR FROM_BINXML_LIB	YYYY-MM-DD hh:mm:ss.nnn XXXXXX_02_00[ppppppppppp]: [E] (UNDEFINED_ERROR) %s %s : Error message.	Error Message as returned by BINXML library.	Error log message generated. Abort L1OP.	Error	No	Yes
9	Problem accessing Object	NRT_ERR_CODE_ERROR ACCESSING_OBJECT	YYYY-MM-DD hh:mm:ss.nnn XXXXXX_02_00[ppppppppppp]: [E] (UNDEFINED_ERROR) %s %s : Error message.	Error Message as returned by algorithms API	Error log message generated. Abort L1OP.	Error	No	Yes
0	Specified Product type is unknown	NRT_ERR_CODE_ERROR PRODUCT_TYPE UNKNOWN	YYYY-MM-DD hh:mm:ss.nnn XXXXXX_02_00[ppppppppppp]: [W] (UNDEFINED_WARNING) File type %s unknown.	Wrong file type	Warning Message as returned by algorithms API	Warning log message generated.	Warning	No
1	No HKTM found	NRT_ERR_CODE_ERROR NO_HKTM INFORMATION	YYYY-MM-DD hh:mm:ss.nnn XXXXXX_02_00[ppppppppppp]: [E] (UNDEFINED_ERROR) %s %s : Error message.	Error Message as returned by algorithms API	Error log message generated. Abort L1OP.	Error	No	Yes

	Error Description	Error Code	Log Message Shape	Variable Content	Processor Reaction			
					Program Behavior	Trace	Exception	Exit program
2	Cache is empty	NRT_ERR_CODE_ERROR CACHE_STILL_EMPTY	YYYY-MM-DD hh:mm:ss.nnn XXXXXX_02_00[ppppppppppp]: [E] (UNDEFINED_ERROR) %s %s : Error message.	Error Message as returned by algorithms API	Error log message generated. Abort L1OP.	Error	No	Yes
3	No CNOISE status found	NRT_ERR_CODE_ERROR NO_CNOISE_STATUS	YYYY-MM-DD hh:mm:ss.nnn XXXXXX_02_00[ppppppppppp]: [W] (UNDEFINED_WARNING) %s %s : Warning message.	Error Message as returned by algorithms API	Warning Message as returned by algorithms API	Warning log message generated.	Warning	No
4	Error when accessing GSL library with invalid argument	NRT_ERR_CODE_ERROR GSL_LIB_INVALID ARGUMENT	YYYY-MM-DD hh:mm:ss.nnn XXXXXX_02_00[ppppppppppp]: [E] (UNDEFINED_ERROR) %s %s : Error message.	Error Message as returned by algorithms API	Error log message generated. Abort L1OP.	Error	No	Yes
5	Product type is not valid.	NRT_ERR_CODE INCORRECT PRODUCT_TYPE	YYYY-MM-DD hh:mm:ss.nnn XXXXXX_02_00[ppppppppppp]: [E] (UNDEFINED_ERROR) File type %s is not valid to read %s file	File type	Error log message generated. Abort L1OP.	Error	No	Yes
6	Null pointer use detected	NRT_ERR_CODE NULL_POINTER	YYYY-MM-DD hh:mm:ss.nnn XXXXXX_02_00[ppppppppppp]: [E] (UNDEFINED_ERROR) Null pointer use.	Null pointer use.	Error log message generated. Abort L1OP.	Error	No	Yes
7	Specified path does not exist or is not accessible	NRT_ERR_CODE ERROR_WRONG_PATH	YYYY-MM-DD hh:mm:ss.nnn XXXXXX_02_00[ppppppppppp]: [E] (UNDEFINED_ERROR) Path %s is not accessible.	Invalid path	Error log message generated. Abort L1OP.	Error	No	Yes
8	Unknown Error returned by DPGS lib.	NRT_ERR_CODE ERROR_FROM_DPGS_LIB	YYYY-MM-DD hh:mm:ss.nnn XXXXXX_02_00[ppppppppppp]: [E] (UNDEFINED_ERROR) %s %s : Error message.	Error Message as returned by DPGS library.	Error log message generated. Abort L1OP.	Error	No	Yes

Table 5: Errors and warnings

The execution of the processor can result in the specific errors and warnings. In this case the "UNDEFINED_ERROR" and "UNDEFINED_WARNING" Error codes from Table 5 are used, with the embedded message.

The processor also generates INFO, and DEBUG messages. The hierarchy used to output the logs is the typically used by log4c produced logs and the one adopted by the DPGS. When setting a log level, the level and the levels above will be outputted according to the following rule:

DEBUG<INFO<WARN<ERROR

Additionally the processors produce special progress messages that will be always outputted, regardless of the log level configuration. These messages are set as [A], ex: [A][100_PRO_000] will indicate the processing has reached 100%.

4.6. Resources and Performances

This section depicts the different resources and performances that shall be expected for the different processors. All performances were measured on a machine with in a equivalent DPGS machine. The scenario for the SCI_L1C correspond to a nominal Full scenario taken from the ESAC back-up.

The machine used for L1OP v620 was selected to closely follow the specifications of the upgraded DPGS servers:

- CPU: Intel® Xeon® Processor E5-2643 (10M Cache, 3.30 GHz, 8.00 GT/s Intel® QPI)
- RAM: 64GB RDIMM, 1600 MHz
- OS: Red Hat Enterprise Linux 6

The number of threads used in the processing was 8.

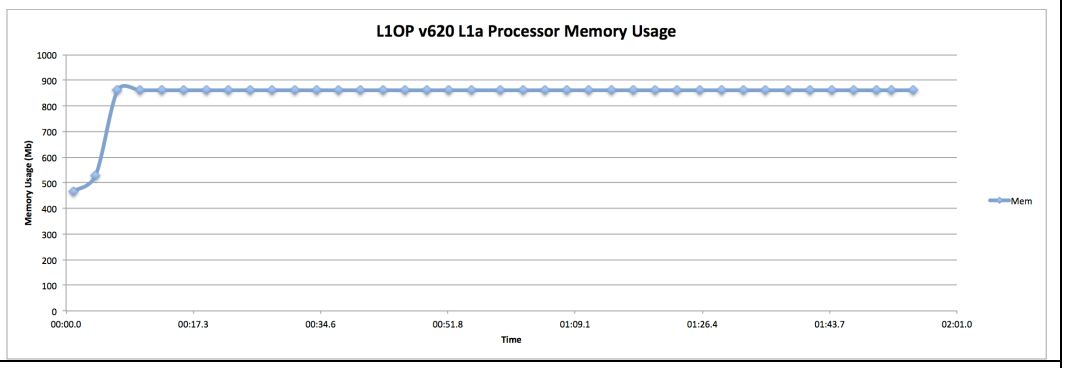
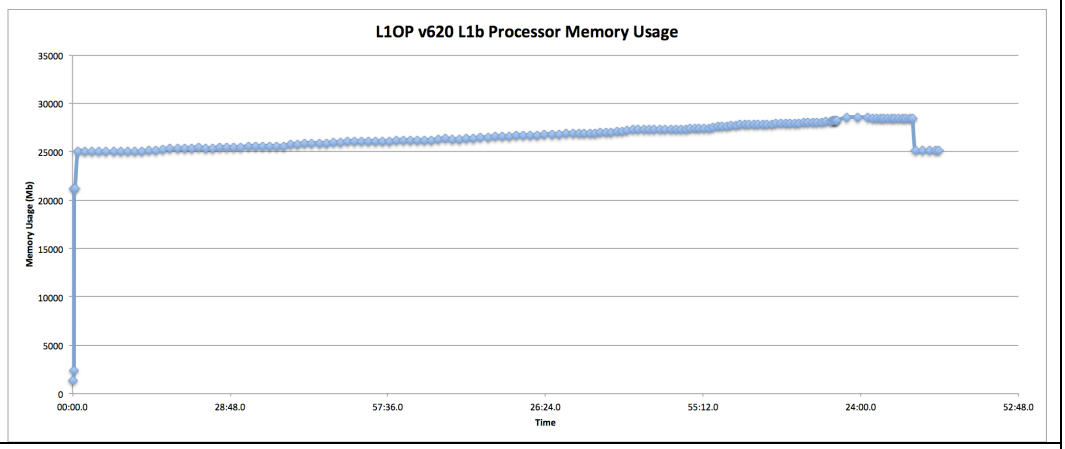
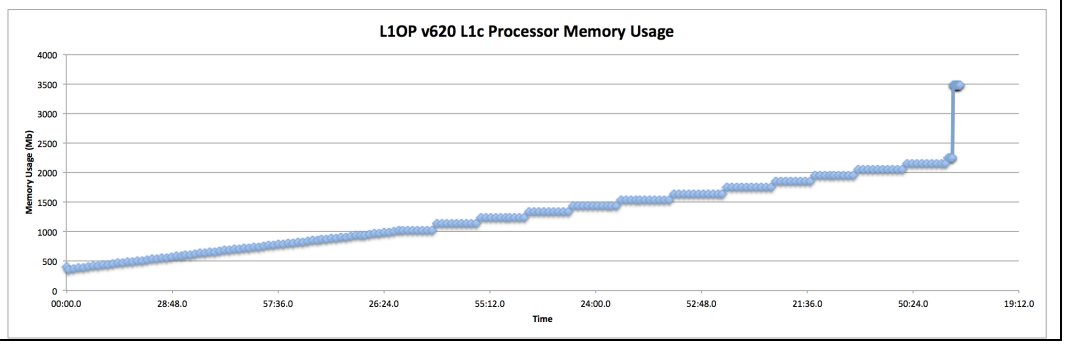
	V600 *(Old DPGS config)*	V620 (Updated DPGS config)	V700 (ALL-LICEF config)
	Execution Time		
NIRCAL	0:07m:18s	00.03m:45s	0:08:32
HKTM1A	0:00:05s	00:00:05s	0:00:05
CAL_1A	0:00:31s	00:00:14s	0:00:21
SCIL1A	0:03m:05s	00:01m:31s	0:01:31
PRL1BF	-	00:24m:14s	0:24:09
SCI_1C	1h:31m:31s	00:31m:47s	0:29:42

Table 6: Execution Times comparison for v600 to v700

As for memory usage, 32 Gb of RAM are recommended, based on PRL1BF requirements. Below is the memory usage of the L1OP v620 for a typical orbit:

Table 7: Memory usage of L1OP v620 for a single orbit

	Memory Usage Consumption

L1A	
L1B	
L1C	

This performance was maintained for the L1OP v700 release.

5. ANNEX 1 - PFW PROCESSOR INSTALLER GENERATION

This section is intended only for the maintainers of the code and not to the code users. The main steps for generating the installer are the following:

- The generation scripts are under $HOME/workspace/`smosnrt/code/commonProcessor/scripts`.
- Run *create_L1OP_pkg.sh* and check the console output for eventual errors. The package is created in the same folder as the creation script is. Package name is like SMOS_<*processor_Name*>_Proc_VV_VV_<centre>_YYYYmmddHHMM.sh. This script takes several options and parameters. If you run the script without parameters or with –h, you'll see them:
  - ⇨ –f <processors information file>. This file contains lines with the following format: "PROCESSOR_NAME:PROCESSOR_VERSION:PROCESSOR_CONFIG_FILE"
  - ⇨ –v <version>. Delivery version number to be delivered (e.g.: 03_50)
  - ⇨ –c <centre>. Processing centre where the software will be installed in. Allowed values are: CEC, FPC and LTA.

This shell script generates the task tables to be used in this version. Before running this script, you must modify the <File_Name> parameter in section <Cfg_File> in every task table file. This parameter points to the processor configuration file.