



Local Processors Re-engineering EPS-SG Processor Prototypes

VII User Manual L1b_LP-MA-XPR-EPS_SG-0209

Version: 2.1
Date: 08/11/2024
Author: Project Team

	<i>Function</i>	<i>Date</i>	<i>Signature</i>
<i>Verified by</i>	Technical Manager	08/11/2024	
<i>Approved by</i>	Project Manager	08/11/2024	

Copyright and confidentiality

This document is the exclusive property of the company Exprivia S.p.A. and cannot be reproduced, even in partial form, without the written authorization of the companies themselves.

Use is limited to all those who, for various reasons, produce company quality management system documentation and / or environmental management system documentation.

Any changes to this document are subject to the provisions of the SGQ and SGQ Documentation Management Procedure (QPRC001S) in the version in force.

Configuration Control

Title: VII User Manual
 Reference File: Formato (Template Generale di Formato ver. 1.01) - ENG.doc
 Created by: Project Team
 Editing Tool: Microsoft WORD

Document Change Record

Ver.	Date	Author/Revision	Reason for Change	Changed Sections/Pages
1.0Draft	20/02/2023	Project Team	First Issue	-
2.0	15/03/2024	Project Team	Issue for AR#1	All
2.1	08/11/2024	Project Team	Issue for review	EUM/EPSSG/AR/4169.9: to check if the usage of variables 'm_max_threads' and 'scan_buffer_size' is properly described in the User Manual Section 6.2.3: table added

Distribution List

Internal Distribution Name	External Distribution Name
Luca Galli	Michela Sunda (EUM)
Stefano Mica	Davide Castellazzi (EUM)
Alessandra Giustiniani	Tomas Esteban (EUM)
Marcello Conte	Carlos Mas
Teresa Geminale	

Table of Content

1. INTRODUCTION	6
1.1. DRD IDENTIFICATION	6
1.2. PURPOSE AND OBJECTIVE	6
2. APPLICABLE AND REFERENCE DOCUMENTS	7
1.1.1. <i>Applicable Documents</i>	7
1.1.2. <i>Reference Documents</i>	7
3. CONVENTIONS	9
4. PURPOSE AND EXTERNAL VIEW OF THE SOFTWARE	10
5. INSTALLATION PROCEDURES.....	11
5.1. HARDWARE AND SOFTWARE CONFIGURATION REQUIREMENTS	11
5.2. LP PACKAGE PRESENTATION	12
5.3. LP PACKAGES INSTALLATION.....	12
1.1.3. <i>Building executable from the source code</i>	12
1.1.4. <i>Building executable from rpm</i>	13
5.4. INSTALLED CONTENT STRUCTURE.....	13
6. OPERATIONS BASICS.....	16
6.1. PROCESSOR EXECUTION	16
6.2. SELECTION AND CONTROL.....	16
6.2.1. INPUT AND VII L0 PRODUCTS	16
6.2.2. ADDITIONAL L0 PRODUCT AND AUXILIARY FILES	16
6.2.3. CONFIGURATION FILES.....	17
6.2.4. JOB ORDER/TASK TABLE	19
6.3. NORMAL OPERATION	19
6.4. ERROR CONDITIONS	19

List of Figures

None

List of Tables

Table 5-1 SW dependencies	12
Table 5-2 Delivery packages	12
Table 6-1 Auxiliary input	17
Table 6-2 VII configuration parameters	19

1. Introduction

This document, the EPS-SG VII L1b LP User Manual (VII-UM) as defined in ECSS-E-ST-40C.

1.1. DRD identification

The Software User Manual (SUM) is called from the normative provisions summarized in Table H-1 of ECSS – ECSS-M-ST-40 (06/03/2009).

This DRD is called from ECSS-E-ST-40 following clauses.

ECSS Standard	Clauses	DRD section
ECSS-E-ST-40	5.5.2.8	All
	5.6.3.3	All
	5.6.4.3	All

1.2. Purpose and Objective

This document is the Software User Manual (SUM) for the EPS-SG VII L1b LP. Its purpose is to provide instructions for the users of the software.

2. Applicable and Reference Documents

1.1.1. Applicable Documents

Document Name	Document Identifier	Internal Reference
EPS-SG LP Project Management Plan	L1b_LP-PL-XPR-EPS_SG-0101 v1.0	[AD-1]
EPS-SG LP Quality Assurance Plan	L1b_LP-PL-XPR-EPS_SG-0202 v1.0	[AD-2]
EPS-SG LP Configuration Status Accounting Report	L1b_LP-RP-XPR-EPS_SG-0102	[CSAR]
Computer Hardware, Operating System and COTS Products Reference Guide	EUM/OPS/TEN/07/2373	[COTS]
EPS-SG Level 0 Product Format Specification	EUM/LEO-EPSSG/SPE/13/70392	[L0-PFS]
EPS-SG VII Level 1B Auxiliary Data Specification	EUM/LEO-EPSSG/SPE/14/777147	[VII-L1B-ADS]
EPS-SG VII Level 1B Product Format Specification	EUM/LEO-EPSSG/SPE/14/777138	[VII -L1B-PFS]
EPS-SG VII Level 1B Product Generation Specification	EUM/LEO-EPSSG/DOC/14/746628	[VII-L1B-PGS]

1.1.2. Reference Documents

Document Name	Document Identifier	Internal Reference
Statement of Work for the Re-engineering of the EPS-SG Processor Prototypes	EUM/LEO-EPSSG/SOW/21/1225668	[RD-1]
EPS-SG Local Mission L1 Processor Technical Requirements	EUM/LEO-EPSSG/DOC/21/1227040	[RD-2]
ICD-VII EPS-SG Local Processor Interface Control Document	L1b_LP-ID-XPR-EPS_SG-0206	[VII-L1B-ICD]
ICD-VII EPS-SG Local Processor Software Design Document	L1b_LP-DD-XPR-EPS_SG-0207	[VII-L1B-SDD]
VII L1B Processor Prototype SW packages	/	[VII-L1B-IPP]

Terms, Definitions and Abbreviated Terms

APT	Anomaly Processing Tool
AR	Acceptance Review
ARB	Anomalies Review Board
BPMN	Business Process Model Notation
BRC	Basic Repeat Cycle
CEFR	Common European Framework of Reference for Languages
DBA	Direct Broadcast Acquisition
CFI	Customer Furnished Item
COTS	Commercial Off-the-Shelf
DFDA	Digital Factory Aerospace & Defence
EO	Earth Observation
EPS-SG	EUMETSAT Polar System-Second Generation
HW	HardWare
ICI	Ice Cloud Imaging
ICD	Interface Control Document
IPR	Intellectual Property Right
KOM	Kick-off Meeting
L1	Level 1
LP	Local Processor
MWI	Microwave Imaging for Precipitation
MWS	Microwave Sounding
NetCDF	Network Common Data Form
NAVATT	NAVigation and ATTitude
NRB	Non-Conformance Review Board
NRT	Near Real Time
N/A	Not applicable
OS	Operating System
OSS	Open Source Software
PFS	Product Format Specification
PGS	Product Generation Specification
PS	Processing Specification
QA	Quality Assurance
RCP	Requirement Check Point
REQ	REquirement
SAF	Satellite Application Facility
SCA	Scatterometer
SOW	Statement of Work
SW	SoftWare
TBC	To Be Confirmed
TBD	To Be Defined
TCE	Technical Computing Environment
TDS	Test Data Set
TN	Technical Note
TRB	Test Review Board
TRR	Test Readiness Review
VCD	Verification Control Document
VII	Visible-Infrared Imaging
WBS	Work Breakdown Structure
WP	Work Package
XPR	Exprivia S.p.A.

3. Conventions

The commands are indicated in *Italic* and framed.

<i>mkdir \$MWS_TMP_DIR</i>

The text to be replaced is surrounded by <>.

<version_number>

4. Purpose and External View of the Software

The **VII L1b LP** SW contains just one main processing chain: the **VII L1b** processor.

The VII L0 input products are processed by the **VII L1b** processor, with the help of the auxiliary files, to generate the corresponding VII L1b products, which are the main output products. In addition, it generates the auxiliary Context Data files, Solar and Engineering auxiliary products.

The definition of the algorithms of the processing chain is given in [VII-L1B-PGS] and [INR-SPEC-SATA]. The definition of the input L0 files is given in [L0-PFS], of the auxiliary files in [VII-L1B-ADS], and of the output L1 files in [VII-L1B-PFS].

The **VII L1b LP** SW is designed to be executed via command line or integrated in a Processing Framework based on job order interface. The Processing Framework will be able to invoke the VII L1b LP processor to generate the output VII L1b products and the auxiliary Context Data files, starting from the inputs data according to the interface defined in [VII-L1B-ICD]. In addition, the VII L1b LP processor can be executed in standalone by command line, with all the input data (products and auxiliary data compliant with EUMETSAT PFSs/ADSs) available in a file system.

5. Installation procedures

This section contains the description of the procedures the operator must use in order to install/uninstall the EPS-SG VII L1b LP .

5.1. Hardware and Software configuration requirements

The installation and execution of the VII L1b LP SW requires to have the following:

System:

- Hardware as defined in [COTS].
- 10 CPU 10Gb RAM 58Gb HDD

Environment:

- Red Hat Enterprise Linux Server release 9.0 Operating System.
- g++ compiler version v11.3.

External libraries:

The list of external libraries needed in compilation and run-time is provided in Table 5 1. The external libraries shall be deployed in the OS prior to the installation of VII L1b LP.

The EO CFI RPM is delivered as an RPM package along with the MWS L1b LP. The other libraries shall be directly deployed using the OS package manager.

SW (COTS, Databases, SU, DU, Test Tools...)	Version	Comment
"VII-L1B-LP-1.1.0.tgz" VII SW source code	1.1.0	
"epssg-VII-L1B-LP-1.1.0-1.el9.x86_64.rpm" VII L1B LP SW executable RPM	1.1.0	
"EOCFI-CPP-4.23-4.23-1.el9.x86_64.rpm " EOCFI Library	4.23	
L0-READER-1.0-1.0-1.el9.x86_64.rpm L0Reader Library	1.0	
NetCDF	4.8.1	
NetCDF-cxx4	4.3.1	
Xerces	3.2.3	
EIGEN	3.4.0	
spdLog	1.10.0	

Table 5-1 SW dependencies

5.2. LP package presentation

The VII L1b LP software is a self-contained delivery that provides the nominal version of the VII L1b LP SW. It consists of two packages, the source code software package [VII-SW] and the rpm installation package [VII-RPM]:

SW package	File	Description
VII L1B Source code	VII-L1B-LP-<x.y.z>.tgz	VII L1 LP processor complete source code, including makefiles
VII L1B Installation package	epssg-VII-L1B-LP-<x.y.z>-1.el9.x86_64.rpm	VII L1 LP processor compiled, ready-to-install RPM executable package for Linux

Table 5-2 Delivery packages

Where x.y.z represents the 3 digits of the SW version number.

5.3. LP packages installation

1.1.3. Building executable from the source code

Starting from the delivered source code compressed package, the user can build the application executable. In order to do so, the user shall follow the next steps:

- Create a local temporary directory to extract and build the application, e.g. "~/my_tmp/mws_inst/", defined by \$VII_TMP_DIR.

```
mkdir $VII_TMP_DIR
```

- Uncompress the software package [VII-SW] into the temporary directory.

```
cd $VII_TMP_DIR
tar xvfz VII-L1B-LP-<x.y.z>.tgz
cd VII-L1B-LP-<x.y.z>
source common_EPSSG/ProjectEnv/epssg.sh
cd components/
make -s abortOnError
cd ../applications
make -s abortOnError
```

Explore the bin folder to list th built executables:

```
ls VII-L1B-LP-1.1.0/bin/
```

1.1.4. Building executable from rpm

The user can install the RPM contents into the system, either the RPM in the delivery package or the one generated by the user from the source code, by running the following:

- Install the executable RPM package.

```
sudo rpm -i epssg-VII-L1B-LP-<x.y.z>-1.el9.x86_64.rpm
```

The previous command installs the **VII L1b LP** executables and libraries in the following path:

`/opt/facilities/epssg/VII-L1B-LP/`

Where x.y.z is the version number of the package, e.g. 1.1.0.

This new directory is automatically linked to the generic directory:

`/opt/facilities/epssg/VII-L1B-LP/`

5.4. Installed content structure

After the software installation is successfully performed, the software executables are available in the directory `/opt/facilities/epssg/VII-L1B-LP/bin`.

The installation package deployment tree is:

```
/opt/facilities/epssg/VII-L1B-LP/
- version.txt
- bin:
  o VII_L1B_LP
- debug
- lib:
  o libAbstractFactoryChain.so.3 -> libAbstractFactoryChain.so.3.7.2
  o libAbstractFactoryChain.so.3.7.2
  o libACS_SSL.so.4 -> libACS_SSL.so.4.2.1
  o libACS_SSL.so.4.2.1
  o libApp.so.13 -> libApp.so.13.0.5
  o libApp.so.13.0.5
  o libConfiguration.so.5 -> libConfiguration.so.5.19.0
  o libConfiguration.so.5.19.0
  o libConvert.so.1 -> libConvert.so.1.7.3
  o libConvert.so.1.7.3
  o libCRC.so.3 -> libCRC.so.3.0.7
  o libCRC.so.3.0.7
  o libDateTime.so.6 -> libDateTime.so.6.0.2
  o libDateTime.so.6.0.2
  o libEPSP_log.so.1 -> libEPSP_log.so.1.0.4
  o libEPSP_log.so.1.0.4
```

- libEPSGVII_common.so.1 -> libEPSGVII_common.so.1.1.3
- libEPSGVII_common.so.1.1.3
- libEPSGVII_inr.so.1 -> libEPSGVII_inr.so.1.2.2
- libEPSGVII_inr.so.1.2.2
- -----|libEPSGVII_navatt.so.1 -> libEPSGVII_navatt.so.1.2.3
- -----|libEPSGVII_navatt.so.1.2.3
- -----|libEPSGVII_products.so.1 ->
- libEPSGVII_products.so.1.0.17
- -----|libEPSGVII_products.so.1.0.17
- -----|libEPSGVII_vii.so.1 -> libEPSGVII_vii.so.1.2.2
- -----|libEPSGVII_vii.so.1.2.2
- -----|libEPSSGJobOrder.so.2 -> libEPSSGJobOrder.so.2.0.0
- -----|libEPSSGJobOrder.so.2.0.0
- -----|libEPSSGStandaloneApp.so.2 ->
- libEPSSGStandaloneApp.so.2.2.0
- -----|libEPSSGStandaloneApp.so.2.2.0
- -----|libException.so.15 -> libException.so.15.2.1
- -----|libException.so.15.2.1
- -----|libFile.so.7 -> libFile.so.7.3.0
- -----|libFile.so.7.3.0
- -----|libJobOrder.so.5 -> libJobOrder.so.5.2.0
- -----|libJobOrder.so.5.2.0
- -----|libLPL1BVIIAuxillaryFunctions.so.3 ->
- libLPL1BVIIAuxillaryFunctions.so.3.0.4
- -----|libLPL1BVIIAuxillaryFunctions.so.3.0.4
- -----|libLPL1BVIIIIngestion.so.1 ->
- libLPL1BVIIIIngestion.so.1.0.6
- -----|libLPL1BVIIIIngestion.so.1.0.6
- -----|libLPL1BVIIIInputDataStructures.so.1 ->
- libLPL1BVIIIInputDataStructures.so.1.0.0
- -----|libLPL1BVIIIInputDataStructures.so.1.0.0
- -----|libLPL1BVIIProcessor.so.1 ->
- libLPL1BVIIProcessor.so.1.0.25
- -----|libLPL1BVIIProcessor.so.1.0.25
- -----|libMQTT.so.6 -> libMQTT.so.6.1.3
- -----|libMQTT.so.6.1.3
- -----|libNetcdfWrapper.so.1 -> libNetcdfWrapper.so.1.33.2
- -----|libNetcdfWrapper.so.1.33.2
- -----|libPlugin.so.5 -> libPlugin.so.5.4.1
- -----|libPlugin.so.5.4.1
- -----|libProc.so.3 -> libProc.so.3.0.5
- -rwxr-xr-x. 1 root root 360552 Mar 15 11:21 libProc.so.3.0.5
- lrwxrwxrwx. 1 root root 18 Mar 15 11:21 libRegExp.so.5 ->
- libRegExp.so.5.8.1
- -rwxr-xr-x. 1 root root 60176 Mar 15 11:21 libRegExp.so.5.8.1
- lrwxrwxrwx. 1 root root 21 Mar 15 11:21 libResources.so.7 ->
- libResources.so.7.0.0
- -rwxr-xr-x. 1 root root 209864 Mar 15 11:21 libResources.so.7.0.0
- lrwxrwxrwx. 1 root root 25 Mar 15 11:21 libSerialization.so.2 ->
- libSerialization.so.2.0.1
- -rwxr-xr-x. 1 root root 460840 Mar 15 11:21 libSerialization.so.2.0.1
- lrwxrwxrwx. 1 root root 18 Mar 15 11:21 libSignal.so.6 ->
- libSignal.so.6.2.2
- -rwxr-xr-x. 1 root root 151424 Mar 15 11:21 libSignal.so.6.2.2

- lrwxrwxrwx. 1 root root 26 Mar 15 11:21 libStandaloneApp.so.5 -> libStandaloneApp.so.5.17.1
- -rwxr-xr-x. 1 root root 164008 Mar 15 11:21 libStandaloneApp.so.5.17.1
- lrwxrwxrwx. 1 root root 30 Mar 15 11:21 libStringFactoryChain.so.6 -> libStringFactoryChain.so.6.1.2
- -rwxr-xr-x. 1 root root 28896 Mar 15 11:21 libStringFactoryChain.so.6.1.2
- lrwxrwxrwx. 1 root root 28 Mar 15 11:21 libStringKeyFactory.so.2 -> libStringKeyFactory.so.2.1.3
- -rwxr-xr-x. 1 root root 17056 Mar 15 11:21 libStringKeyFactory.so.2.1.3
- lrwxrwxrwx. 1 root root 24 Mar 15 11:21 libStringUtils.so.1 -> libStringUtils.so.1.49.0
- -rwxr-xr-x. 1 root root 107992 Mar 15 11:21 libStringUtils.so.1.49.0
- lrwxrwxrwx. 1 root root 16 Mar 15 11:21 libTask.so.3 -> libTask.so.3.0.0
- -rwxr-xr-x. 1 root root 385136 Mar 15 11:21 libTask.so.3.0.0
- lrwxrwxrwx. 1 root root 22 Mar 15 11:21 libTaskTable.so.5 -> libTaskTable.so.5.14.0
- -rwxr-xr-x. 1 root root 653280 Mar 15 11:21 libTaskTable.so.5.14.0
- lrwxrwxrwx. 1 root root 18 Mar 15 11:21 libThread.so.7 -> libThread.so.7.0.1
- -rwxr-xr-x. 1 root root 164344 Mar 15 11:21 libThread.so.7.0.1
- lrwxrwxrwx. 1 root root 18 Mar 15 11:21 libTimer.so.2 -> libTimer.so.2.19.3
- -rwxr-xr-x. 1 root root 35440 Mar 15 11:21 libTimer.so.2.19.3
- lrwxrwxrwx. 1 root root 20 Mar 15 11:21 libTIOOrder.so.2 -> libTIOOrder.so.2.18.2
- -rwxr-xr-x. 1 root root 144664 Mar 15 11:21 libTIOOrder.so.2.18.2
- lrwxrwxrwx. 1 root root 25 Mar 15 11:21 libXMLResources.so.5 -> libXMLResources.so.5.74.7
- -rwxr-xr-x. 1 root root 1075952 Mar 15 11:21 libXMLResources.so.5.74.7

The VII L1b LP processor executable has the following name:

"VII_L1B_LP"

6. Operations Basics

The SUM shall define the operational tasks, identifying their sequence and hierarchy, the roles and the staffing, the standard daily operations and the contingency operations.

6.1. Processor Execution

The **VII L1b LP** processor is executed launching a binary, using as only input a job order. This job order can be automatically generated by the Processing Framework or can be written by the user following [VII-L1B-ICD].

In both cases, the interface executable is started by a single command:

```
<executable_name> <joborder_file.xml>
```

Where:

- *executable_name* is the name of the executable to be run, i.e., /opt/facilities/epssg/VII-L1B-LP/VII_L1B_LP
- *joborder_file.xml* is the XML file containing the input/output files information, as defined in [VII-L1B-ICD].

6.2. Selection and control

6.2.1. Input and VII L0 products

The VII L1b LP is designed to process one or several VII L0 file(s), which cover a time-period equal or greater than a minimum time span. As per local processor requirement, any length equal or greater than 2.5 minutes can be processed; the actual minimum length of input data depends on the several processing averaging windows defined in the auxiliary input calibration file.

In the context of local processing nominal conditions, assuming that L0 data do not overlap, a sequential processing of the VII L0 files is mandatory, and the reception of VII L0 product is triggering the processing.

The two VII L0 granules surrounding the triggering VII L0 product are also passed as input to ensure nominal processing at the edges of the triggering product, if they are available. The time interval [T1, T2] to process, defined by the start sensing time and stop sensing time parameters in the Job Order, corresponds to the start and stop of the triggering VII L0 product.

The output of the VII L1B LP will be a product spanning the time interval [T1, T2] continuous in terms of content, metadata, and quality, with no degradations and discontinuities at the boundaries.

6.2.2. Additional L0 product and Auxiliary files

From VII L1b LP utilisation point of view, all input files other than the VII L0 products can be considered as Auxiliary data (static or dynamic). These files are listed in **Errore. L'origine riferimento non è stata trovata.** and deeper described in [VII-L1B-ICD].

When several types of files can be used, the priority is indicated in the column "Priority".

Note that, according to the VII L1b LP Task Table, the Processing Framework will select:

- All VII L0 and NAVATT L0 files that cover partly the time interval specified in the Job Order (ValIntersect selection rule).
- The Static auxiliary files with the latest creation date (LatestValCover selection rule), for each type of file.

File_Type	Description	Priority
NAV-00-SRC	NAVATT Level 0 product. It contains SGA Navigation and Attitude information generated by the Satellite	Orbit: 1 st choice
_____AUX_POFD	The predicted orbit file contains orbit state vectors for a given Metop-SG satellite with a time step of 1 minute and in Earth-Fixed reference frame	Orbit: 2 nd choice Time correlation: 1 st choice
_____AUX_IBA_	The IERS bulletin A provides information on Earth's orientation in the IERS Reference System	Time correlation: 2 nd choice
VII_1B_AUX_CCDB	The VII L1B Processing Parameters data files contain all required auxiliary information needed by the VII L1B processor in NRT processing	
VII_1B_AUX_LMDB	The VII L1B landmark database contains the landmarks for the level 1 processor.	
_____AUX_RD09	Raster Digital Elevation Model (9 arc-seconds). Provides surface elevations at 9 arc-seconds spatial resolution.	
_____AUX_LSM_	MWS Land Sea Mask. Provides information on whether the pixel is above land, sea or coast.	
VII_1B_AUX_SOL_	The VII L1B Solar calibration co-efficient data files contain the gain calibration coefficients for the solar calibration which is updated once per day over the South Pole.	

Table 6-1 Auxiliary input

6.2.3. Configuration files

The **VII L1b LP** processor is configurable by means of an external auxiliary data file, the VII L1B Configuration and Calibration Parameters package, which is specified in [VII-L1B-ICD] and in [VII-L1B-ADS]. The user can modify the configuration parameters in that file and, without the need of re-compiling the software, the processor will have a different behavior. These configuration parameters are typically switches and selectors.

The VII L1B Configuration and Calibration Parameters package is composed of 7 files:

- **manifest.xml**: the manifest describing the package content, which format and content are defined in [VII-L1B-ADS], updated to include the additional configuration files for VII L1b LP SW.
- **B01.xml**: This xml files contains the format specification of the breakpoint 1
- **B02.xml**: This xml files contains the format specification of the breakpoint 2
- **B03.xml**: This xml files contains the format specification of the breakpoint 3
- **B04.xml**: This xml files contains the format specification of the breakpoint 4
- **ENG.xml**: This xml files contains the format specification of the Engineering Product
- **RAD.xml**: This xml files contains the format specification of the VII L1B Product
- **SOL.xml**: This xml files contains the format specification of the Solar Product
- **rsync01001.nc**: This file contains the Airbus D&S OSS Geometry Simulation
- **VII_L1B_CNF__20230727.nc**: Configuration Parameter for the VII Processors

- VIIPackets.xml: Level 0 packed structure definition

The configuration parameters which are intended to be modified by the user for its specific needs are gathered in the VII L1b LP configuration parameters file (VII_1B_AUX_CNF_). List of parameters described in table

Most of the configuration parameters stored inside the other files of the VII L1B Configuration and Calibration Parameters package are static and frozen unless the VII instrument or [VII-L1B-PGS] specifications evolve.

Configuration parameter definition

Name	Description	Value
aux_GR	generation context global/regional Possible values [GR_]	108.42
aux_class	Type of activity for which the file is used OPER TEST TD[0-9][0-9] SVT[0-9]	OPER
aux_environment	Environment Identifier OPE VAL IVV DEV ENG	OPE
channel_mask	Can be used to avoid processing of certain channels.	0
deflate_level	Netcdf compression parameter.	0
chunking_size_act	Netcdf compression parameter.	0
chunking_size_alt	Netcdf compression parameter.	0
disposition_mode	disposition_mode [TCOV]	C
max_threads	The maximum number of parallel threads the processor should launch.	4
mission_type	mission_type [GRL]	R
packet_buffer_size	The maximum number of packets which should be kept in memory.	1000
processing_mode	processing_mode [NBR]	B
scan_buffer_size	The maximum number of scans which should be kept in memory.	5
shuffle	Netcdf compression parameter.	0

Table 6-2 VII configuration parameters

Parallelization is done by processing scans in parallel. The maximum number of threads run in parallel can therefore be limited by `scan_buffer_size` as well as by the total number of scans to be processed. The optimal `scan_buffer_size` is usually $N * \text{max_threads} + 1$ with integer N .

6.2.4. Job Order/Task Table

The Job Order is written in ASCII-based XML-format specified in [VII-L1B-ICD] and in [L1-AD].

The parameters read from the Job Order files are dynamical ones, in the sense that each time the Job Order is generated by the Processing Framework, it can be different. When using the Processing Framework, the Job Order is filled automatically from the information available in the Task Table. Alternatively, if the processor is executed manually at prompt level, it can be fed with a hand-made Job Order file.

6.3. Normal Operation

In the nominal case, the processing ends successfully with return code set to 0 (success).

6.4. Error Conditions

In case of ending with an error, the processing terminates with a return code set to a value between 128 and 255. Messages of level ERROR are written in the log file to explain the root cause of the failure.

Error and Warning messages output by the **MWS L1b LP** are listed in [VII-L1B-ICD] section 12.

End of Document