


	Document - No. <b>AE-PL-KNMI- L2BP-001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>1/ 62</b>	
	Doc. - Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				

<u>Doc. - No.:</u>	<b>AE- PL- KNMI- L2BP- 001</b>
<u>Doc. - Title:</u>	<b>Aeolus Level 2b Processor Software Verification &amp; Validation Plan</b>
<u>Number of pages:</u>	62 pages
<u>Prepared by:</u>	Jos de Kloe (KNMI), David Tan (ECMWF), Dorit Huber (DLR)





	Document- No. <b>AE- PL- KNMI- L2BP- 001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>2/ 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				

## Document Change Log

Issue.	Date	Modified pages	Observations	Name
V 1.0	15- Dec- 2006	--	first (draft) version	J. de Kloe/D. Huber
V1.1	30- Jan- 2007	almost every page	ESA comments of 15- Jan- 2007 incorporated	J. de Kloe
V1.2	14- Feb- 2007	See change bars	ESA comments of 31- Jan- 2007 incorporated	J. de Kloe
<u>V1.3</u>	<u>1123-</u> <u>AprMar-</u> 2007	<u>See change bars</u>	<u>Adapted to updated SRD v1.1 dated 23- Feb- 2007, and</u> <u>adapted to updated code release 1.3</u>	<u>J. de Kloe</u>





	Document- No. <b>AE- PL- KNMI- L2BP- 001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>3/ 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				

## Table of Contents



Front Page.....	1
Document Change Log.....	2
Table of Contents .....	3
1 Introduction and Purpose of Document .....	8
2 Documents.....	9
2.1 Applicable Documents .....	9
2.1 Reference Documents .....	9
2.2 Variable names.....	10
2.3 Acronyms .....	10
3 Test Procedure Overview.....	12
3.1 Test Items.....	12
3.2 Acceptance Test Cases.....	12
3.3 Test Data Summary.....	14
4 Acceptance Test Procedures.....	16
4.1 Test L2bP_DEL_01.....	16
4.1.1 Purpose.....	16
4.1.2 Prerequisites.....	16
4.1.3 Input Specification .....	16
4.1.4 Specific Test Procedure .....	16
4.1.5 Output Specification .....	18
4.1.6 Pass/fail criteria.....	18
4.2 Test L2bP_GEN_01.....	18
4.2.1 Purpose.....	18
4.2.2 Prerequisites.....	18
4.2.3 Input Specification .....	18
4.2.4 Output Specification .....	18
4.2.5 Specific Test Procedure.....	18
4.2.6 Pass/fail criteria.....	19
4.3 Test L2bP_GEN_02.....	19
4.3.1 Purpose.....	19
4.3.2 Prerequisites.....	19
4.3.3 Input Specification .....	19
4.3.4 Output Specification .....	19
4.3.5 Specific Test Procedure .....	20
4.3.6 Pass/fail criteria.....	21



	Document- No.	Issue:	Date:	Page:	
	AE- PL- KNMI- L2BP- 001	V 1.32	11.04.2007	4/ 62	
	Doc.- Title:				
	Aeolus Level 2B Processor Software Verification & Validation Plan				



4.4 Test L2bP_GEN_03.....	21
4.4.1 Purpose.....	21
4.4.2 Prerequisites.....	21
4.4.3 Input Specification .....	21
4.4.4 Output Specification .....	21
4.4.5 Specific Test Procedure.....	21
4.4.6 Pass/fail criteria.....	22
4.5 Test L2bP_GEN_04.....	22
4.5.1 Purpose.....	23
4.5.2 Prerequisites.....	23
4.5.3 Input Specification .....	23
4.5.4 Output Specification .....	23
4.5.5 Specific Test Procedure.....	23
4.5.6 Pass/fail criteria.....	24
4.6 Test L2bP_GEN_05.....	24
4.6.1 Purpose.....	24
4.6.2 Prerequisites.....	24
4.6.3 Input Specification .....	24
4.6.4 Specific Test Procedure.....	24
4.6.5 Output Specification .....	25
4.6.6 Pass-fail criteria.....	25
4.7 Test L2bP_HMI_01.....	25
4.7.1 Purpose.....	25
4.7.2 Prerequisites.....	25
4.7.3 Input Specification .....	25
4.7.4 Output Specification .....	25
4.7.5 Specific Test Procedure.....	25
4.7.6 Pass/fail criteria.....	28
4.8 Test L2bP_EXT_01.....	28
4.8.1 Purpose.....	28
4.8.2 Prerequisites.....	28
4.8.3 Input Specification .....	28
4.8.4 Output Specification .....	28
4.8.5 Specific Test Procedure.....	28
4.8.6 Pass/fail criteria.....	29
4.9 Test L2bP_EXT_02.....	29
4.9.1 Purpose.....	29
4.9.2 Prerequisites.....	29



	Document- No. <b>AE- PL- KNMI- L2BP- 001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>5/ 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				



4.9.3 Input Specification .....	29
4.9.4 Output Specification .....	29
4.9.5 Specific Test Procedure .....	30
4.9.6 Pass/fail criteria .....	30
4.10 Test L2bP_EXT_03.....	30
4.10.1 Purpose.....	31
4.10.2 Prerequisites.....	31
4.10.3 Input Specification .....	31
4.10.4 Output Specification .....	31
4.10.5 Specific Test Procedure .....	31
4.10.6 Pass/fail criteria .....	32
4.11 Test L2bP_PROD_01.....	32
4.11.1 Purpose.....	32
4.11.2 Prerequisites.....	33
4.11.3 Input Specification .....	33
4.11.4 Output Specification .....	33
4.11.5 Specific Test Procedure .....	33
4.11.6 Pass/fail criteria .....	34
4.12 Test L2bP_PROD_02.....	35
4.12.1 Purpose.....	35
4.12.2 Prerequisites.....	35
4.12.3 Input Specification .....	35
4.12.4 Output Specification .....	35
4.12.5 Specific Test Procedure .....	35
4.12.6 Pass/fail criteria .....	36
4.13 Test L2bP_PROD_03.....	36
4.13.1 Purpose.....	36
4.13.2 Prerequisites.....	36
4.13.3 Input Specification .....	37
4.13.4 Output Specification .....	37
4.13.5 Specific Test Procedure .....	37
4.13.6 Pass/fail criteria .....	37
4.14 Test L2bP_PROD_04.....	37
4.14.1 Purpose.....	37
4.14.2 Prerequisites.....	37
4.14.3 Input Specification .....	37
4.14.4 Output Specification .....	38
4.14.5 Specific Test Procedure .....	38



	Document- No.	Issue:	Date:	Page:	
	AE- PL- KNMI- L2BP- 001	V 1.32	11.04.2007	6/ 62	
	Doc.- Title:				
	Aeolus Level 2B Processor Software Verification & Validation Plan				



4.14.6 Pass/fail criteria.....	38
4.15 Test L2bP_PERF_01.....	39
4.15.1 Purpose.....	39
4.15.2 Prerequisites.....	39
4.15.3 Input Specification .....	39
4.15.4 Output Specification .....	39
4.15.5 Specific Test Procedure .....	39
4.15.6 Pass/fail criteria .....	39
4.16 Test L2bP_TOOLS_01.....	40
4.16.1 Purpose.....	40
4.16.2 Prerequisites.....	40
4.16.3 Input Specification .....	40
4.16.4 Output Specification .....	40
4.16.5 Specific Test Procedure .....	40
4.16.6 Pass/fail criteria.....	41
4.17 Test L2bP_TOOLS_02.....	41
4.17.1 Purpose.....	41
4.17.2 Prerequisites.....	41
4.17.3 Input Specification .....	41
4.17.4 Output Specification .....	41
4.17.5 Specific Test Procedure .....	41
4.17.6 Pass/fail criteria.....	42
4.18 Test L2bP_TOOLS_03.....	42
4.18.1 Purpose.....	42
4.18.2 Prerequisites.....	42
4.18.3 Input Specification .....	42
4.18.4 Output Specification .....	42
4.18.5 Specific Test Procedure .....	42
4.18.6 Pass/fail criteria.....	43
4.19 Test L2bP_TOOLS_04.....	43
4.19.1 Purpose.....	44
4.19.2 Prerequisites.....	44
4.19.3 Input Specification .....	44
4.19.4 Output Specification .....	44
4.19.5 Specific Test Procedure .....	44
4.19.6 Pass/fail criteria.....	46
A Verification Cross- Reference Matrix .....	47
B Test Tools.....	60



	Document- No.	Issue:	Date:	Page:	
	AE-PL-KNMI- L2BP-001	V 1.32	11.04.2007	7/ 62	
	Doc.- Title:				
	Aeolus Level 2B Processor Software Verification & Validation Plan				

B.1 unix tools.....	60
B.2 external tools.....	60
B.3 custom tools.....	60
B.4 matlab tools.....	61



	Document- No.	Issue:	Date:	Page:	
	AE- PL- KNMI- L2BP- 001	V 1.32	11.04.2007	8/ 62	
	Doc.- Title:				
	Aeolus Level 2B Processor Software Verification & Validation Plan				

## 1 Introduction and Purpose of Document

The present document is the Software Verification & Validation Plan for the ADM L2B operational processor ~~-, and the operational Level- 1b EE2BUFR convertor. It is part of the deliverables of the contract study "Aeolus ...." and one of the outputs of WP.....~~



This document describes the tests to be conducted in order to verify that the requirements stated for the Operational L2B processor and the L1B EE2BUFR convertor as listed in [AD 7] are met.

Chapter 2 contains the list of applicable and referenced documents and a list of acronyms, and chapter 3 gives an overview of the testing procedure. Chapter 4 gives details of all tests defined in this plan. In appendix A on page 47 the cross-reference matrix is given, and finally in appendix B on page 61 a list is given of external tools, needed by this plan.

### Todo's:

- David Tan suggested adding a new chapter to describe all scientific tests in this document as well.
- generate the needed TDS files defined in Table 3.3 using the newest E2S and L1BP



	Document- No. <b>AE- PL- KNMI- L2BP- 001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>9/ 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				

## 2 Documents

### 2.1 Applicable Documents

**[AD 1]** PDS-IPF ICD Generic Interface Guideline, Issue 2.2, 01/08/2006, **ESA-ID- ACS-GS-0001**

**[AD 2]** Statement of Work: 'Aeolus Level 2B/2C Processing Facility', Issue 1B,16/09/2004, **AE-SW-ESA-GS-0117**, Supplemented by [AD 3]

**[AD 3]** Statement of Work: 'Enhanced Aeolus Level 2B/2C Functionalities & Pre-Launch Validation', Issue 1.0, 29/11/2005, **AE-SW-ESA-GS-023**

**[AD 4]** ADM\_Aeolus Implementation of Level 2B/2C Processing Facility - Technical Requirements, Issue 1.1, 06/06/2004, **XADM-GSEG-EOPG-RD-04-0003**

**[AD 5]** ADM-Aeolus PCD-ECMWF Interface Control Document, Issue 1.5, 18/10/2006, **XADM-GSEG-EOPG-ID-04-0002**

**[AD 6]** Earth Explorer ground Segment File Format Standard, Issue 1.4, 13/06/2003, **PE-TN-ESA-GS-0001**

**[AD 7]** Aeolus Level 2B Processor Software Requirements Document, Issue 1.~~10~~, ~~2304/0212/2007~~~~6~~, **AE-RS-ECMWF-L2BP-001**

**[AD 8]** ADM-Aeolus Level 2B/2C Processor input/output data definitions interface control document, issue 1.~~32~~, ~~2311/0212/2007~~~~6~~, **AE-IF-ECMWF-L2BP-001** (~~previously labelled as [RD4]~~)

**[AD 9]** Aeolus Level 2B Processor External Interface control document, issue 1.0, 11/12/2006, **AE-IF-ECMWF-L2BP-002** (~~previously labelled as [RD5]~~)

**[AD 10]** WMO FM94 (BUFR) description of ADM-Aeolus L1B/L2B products, Issue 1.0, 06/03/2006, **AE-TN-ECMWF-L2P-0072-TEMPLATE**

**[AD 11]** ADM-Aeolus Level- 1B EE2BUFR Converter: Installation and User Instruction, Issue 1.0, 11/12/2006, **AE-TN-ECMWF-L2BP-0072**

### 2.1 Reference Documents



**[RD 1]** Aeolus Level 2B Processor Design Document. Currently referred to as: Definition of Baseline Aeolus Level 2B Processing and Design, Issue 1.2, 09/09/2005, **AE-TN-ECMWF-L2BP-0022**

**[RD 2]** Level 1B Input/Output data definitions interface control document, issue 3.0, 31/07/2006, **ADM-IC-52-1666**

**[RD 3]** Aeolus Level 2A Processor Input/Output data definition, issue 1.3, ~~1704/0109/2007~~~~6~~, **AE-IF-DLR-L2A-004**

**[RD 4]** Aeolus Level 2B Algorithm theoretical baseline document. Currently referred to as: Mathematical description of the Aeolus Level 2B Processor, Issue 2.~~10~~, ~~2305/042/2007~~~~6~~, **AE-TN-ECMWF-L2BP-0023**



	Document- No. <b>AE- PL- KNMI- L2BP- 001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>10/ 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				

**[RD 5]** Aeolus Level- 2B Processor Software User Manual, issue 1.2, 07/12/2006, **AE- MA- ECMWF- L2BP- 002**

~~**[RD6]** Software Maintenance Plan, to be written? [note by Herbert: Warranty and maintenance should be defined in the "Software and Maintenance Plan" (SAF: maintenance folder), see: SOW, iss 1B, table A1.5- 1. (for this processor version, the description of warranty + maintenance could be included in the release note.)]~~

## 2.2 Variable names

In this section some names are defined, that are used in several places in this document, and which change for each new software delivery. The definition of these names in this place makes it possible to refer to them in the remainder of this document, which makes maintenance of this document much easier.



<i><b>variable name</b></i>	<i><b>variable definition</b></i>
L2BP_source	L2BP_Release1. <u>32A</u> _source.tar.gz
L2BP_datapack	L2BP_Release1. <u>32A</u> _datapack_200 <u>7041061218</u> .tar.gz
L2BP_TDS	L2BP_Release1. <u>32A</u> _TDS.tar.gz [note that this file is not yet available]

Table 2.1.: definitions of some variables

## 2.3 Acronyms



ADM- Aeolus	Atmospheric Dynamics Mission: Aeolus keeper of the winds
ATBD	Algorithm Theoretical Baseline Document
BUFR	Binary Universal Form for Representation (of data)
CLI	Command Line Interface
COTS	Commercial Off- The- <u>Shelf</u> (software)
EE	Earth Explorer
EE2BUFRGuide	Earth- Explorer to BUFR conversion tool guide
ESA	European Space Agency
ESTEC	European Space Research and Technology Centre
ExtICD	External Interface Control Document
FAT	Factory Acceptance Test
<u>FEP</u>	<u>Front- End- Processor</u>
FWHM	Full Width Half Maximum
HLOS	Horizontal Line- Of- Sight
ICD	Interface Control Document
IODD	Input/Output Definitions Document
<u>L2BP</u>	<u>Level 2B Processor</u>
LOS	Line- Of- Sight



	Document- No.	Issue:	Date:	Page:	
	AE- PL- KNMI- L2BP- 001	V 1.32	11.04.2007	11/ 62	
	Doc.- Title:				
	Aeolus Level 2B Processor Software Verification & Validation Plan				

<u>LTA</u>	<u>Long Term Archive</u>
<u>MPH</u>	<u>Main Product Header</u>
<u>OS</u>	<u>Operating System</u>
RMA	Reference Model for the Atmosphere
SNR	Signal- to- Noise Ratio
<u>SPH</u>	<u>Specific Product Header</u>
SRD	Software Release Document
SRN	Software Release Notes
SUM	Software User Manual
TDS	Test Data Set
TLE	Thin Layer Emulator
VCRM	Verification Cross Reference Matrix



	Document- No. <b>AE- PL- KNMI- L2BP- 001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>12/ 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				

## 3 Test Procedure Overview

### 3.1 Test Items

The item under test during the Factory Acceptance Testing is the L2bP system as designed and built by ECMWF and subcontractors. The specific L2bP requirements that will be verified during factory acceptance testing (FAT) are identified in [AD 7].

All deliverable software, many ascii data files and system build files will be under the control of Configuration Management (PerForce). Binary data files, and some associated ascii header files will be combined in a binary datapack. Each new version of this datapack will be archived in the ECMWF tape- archive (ecfs).

The system software build and install procedures will be conducted at the start of the FAT, as part of the Factory Acceptance Test Procedures. If changes are required to the software between FAT events, the system software will be rebuilt again from the original copy retrieved from the version control system (which will be made available using the eRoom).

All required vendor- supplied software; such as the operating system, compilers, and external libraries will be installed and configured prior to the tests. Test data sets, if required, will be identified in this document. They will be made available as binary datapack and archived at ECMWF to ensure the repeatability of the Acceptance Tests.

The test procedures in this document can be used for regression testing of future versions of the L2bP software after the initial Factory Acceptance Test.

### 3.2 Acceptance Test Cases

Table 3.1 below summarises the main test classes to be performed on the L2bP:

Test Case	Test Case Identifier	Description
L2bP Delivery	L2bP_DEL	Includes a validation checklist for L2bP deliverables.
L2bP General	L2bP_GEN	Includes software compilation and installation, validation of Bit/Byte, documentation, maintenance, reliability, and security requirements.
L2bP HMI	L2bP_HMI	Includes validation of the commandline usage of the main processor, and command line tools that can be used for construction of auxiliary datafiles.
L2bP External Interface	L2bP_EXT	Includes validation of the Job Order, Products List, event logging, exit code, processor control, and product report interface requirements, as required by the ThinLayer. (Also requirements needed for the standalone and IFS integrated versions of the L2bP can be stated here)
L2bP Product Generation and validation	L2bP_PROD	Includes validation of Level 2b processing requirements, event logging, data validation, processing reference frames, Product Reports, and validation of Level 2b product requirements.



Test Case	Test Case Identifier	Description
L2bP Performance	L2bP_PERF	Includes validation of L2bP performance requirements.
L2bP Tools	L2bP_TOOLS	Includes testing of additional tools delivered. (this covers the L1B EE2BUFR conversion tool)
L2bP Science	L2bP_Science	includes all tests needed to verify the correct scientific working of the L2bP software

**Table 3.1 : L2bP Acceptance Test Cases**

Table 3.2 below gives the full list of tests to be performed on the L2bP:

<i>Test Case name</i>	<i>description</i>	<i>see section for full details</i>
L2bP_DEL_01	check that all needed documentation and data files are delivered	4.1
L2bP_GEN_01	check the hardware	4.2
L2bP_GEN_02	check the formatting and contents of the documentation. check the source code of the software.	4.3
L2bP_GEN_03	installation of the software on a linux platform	4.4
L2bP_GEN_04	handling of mathematical exceptions	4.5
L2bP_GEN_05	check bit and byte ordering	4.6
L2bP_HMI_01	general check of the correct functioning of the processor and the CLI	4.7
L2bP_EXT_01	check the external (Thin Layer Emulator, TLE) interface	4.8
L2bP_EXT_02	check the documentation and example files needed for using the TLE	4.9
L2bP_EXT_03	check generation of warning, error and other log messages	4.10
L2bP_PROD_01	Test generation of a product report according to specs	4.11
L2bP_PROD_02	Test the capability to handle files containing missing or corrupted input numbers, or missing input files	4.12
L2bP_PROD_03	Test the L2bP capability to process a very large L1B inputfile, containing 8 orbits worth of data	4.13



<i>Test Case name</i>	<i>description</i>	<i>see section for full details</i>
L2bP_PROD_04	Test the formatting of the L2B product	4.14
L2bP_PERF_01	Test the runtime performance of the L2BP when processing 1 full orbit worth of data	4.15
L2bP_TOOLS_01	demonstrate by running the TestRead* tools in each of the file handling directories available for handling each auxiliary file.	4.16
L2bP_TOOLS_02	test any tools for graphical display of auxiliary (calibration) data files	4.17
L2bP_TOOLS_03	Test generation and/or editing of auxiliary input files	4.18
L2bP_TOOLS_04	Demonstrate the EE2BUFR tool	4.19
L2bP_SCIENCE_01	verify the correct scientific working of the L2bP software. These tests are described in TN 3.1 for now (RD..) and will not be repeated in detail here.	

*Table 3.2: full list of test definitions*

### 3.3 Test Data Summary



Table 3.3 summarizes the test data sets used in the various Test Procedures. These data sets are created using the MDA E2S and MDA L1b Processor and different versions are archived in the ECMWF ecfs system. All tests not mentioned in Table 3.3 below use datafiles provided with the default L2B processor source code and datapack.

[todo: check if this is sufficient for us]

[todo: these files still need to be created!]



<b>TDS ID</b>	<b>Duration</b>	<b>Notes</b>	<b>Used by Job Order</b>	<b>Used for Test</b>
L2B_TDS_01	1 orbit (~200 BRCs)	used to test the runtime performance for processing a typical L1B product	tbd	L2bP_PERF_01
L2B_TDS_02	2 BRC's (~0.01 orbit)	file containing missing and/or corrupted data	tbd	L2bP_PROD_02
L2B_TDS_03	8 orbits (~1600 BRC's)	used to test the ability to process an extremely large L1B product	tbd	L2bP_PROD_03



	Document- No.	Issue:	Date:	Page:	
	AE-PL-KNMI-L2BP-001	V 1.32	11.04.2007	15 / 62	
	Doc.- Title:				
<b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>					

**Table 3.3: L2bP Acceptance Test Data Set Summary**



	Document- No. <b>AE- PL- KNMI- L2BP- 001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>16 / 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				

## 4 Acceptance Test Procedures

This chapter describes all tests needed to demonstrate that all requirements are met. The full list of tests is given in Table 3.2. The tracability matrix defining which test demonstrates which requirement is given in Table 4.21 in appendix A on page 47.

In the next subsections further details will be given on each defined test, on the tests purpose, prerequisites, input and output specifications, specific test procedure, and expected results (pass-fail criteria).

Note that when the notation \$(build) is used the tester should not literally copy the string \$(build) to the commandline. Rather he should replace the variable name by the appropriate (directory) name. It is strongly advised to use absolute pathnames when doing this.

**Note that some tests should be completed before some other tests are started. The recommended order of execution of tests is the order specified in this document.**

### 4.1 Test L2bP\_DEL\_01

This test procedure verifies delivery requirements for the L2bP software.

#### 4.1.1 Purpose

Test L2bP\_DEL\_01 verifies requirements: [L2BP 140], [L2BP 150], [L2BP 160], [L2BP 170]

#### 4.1.2 Prerequisites

The L2bP software and documentation must have been copied on the target platform.

#### 4.1.3 Input Specification



All commands are entered by the operator via Linux command line.

#### 4.1.4 Specific Test Procedure

Commands to be executed are:

<i><b>Procedure / Command</b></i>	<i><b>comments/ Observations</b></i>	<i><b>passed/ failed</b></i>
tar ztvf L2BP_Release1.32A_TDS.tar.gz		
Verify [L2BP 140] by observing that the datafiles needed for the FAT as mentioned in Table 3.3, named L2B_TDS_01, etc. are present in the listing produced by the above tar command.		
tar ztvf L2BP_Release1.32A_source.tar.gz		
Verify that an example of the AUX_PAR_2B file named: ThinLayer/AE_TEST_AUX_PAR_2B_20050331T000000_20111231T000000_0000.EEF is present in the listing produced by the above tar command.		



	Document- No. <b>AE- PL- KNMI- L2BP- 001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>17/ 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				

<i>Procedure / Command</i>	<i>comments/ Observations</i>	<i>passed/ failed</i>
tar ztvf L2BP_Release1.32A_datapack_2007041061218.tar.gz		
Verify that an example of the AUX_MET_12 file named: Test/AMD_file_handling/AE_TEST_AUX_MET_12_20041 201T000000_20041201T003000_0001.DBL.expected.L INUX_pg90  is present in the listing produced by the above tar command		
Verify that an example of the AUX_RBC file named: RayleighBrillouinProcessing/AE_TEST_AUX_RBC_AX_200 60703T123456_20070703T123456_0001.DBL  is present in the listing produced by the above tar command		
Verify that an example of the AUX_CLM file named: Test/AuxClim_file_handling/AE_TEST_AUX_CLM_AX_20 010312T123456_20110312T123456_0001.DBL.expect ed.LINUX_pg90  is present in the listing produced by the above tar command		
Verify that an example of the AUX_CAL file named: [an example has not yet been generated]  is present in the listing produced by the above tar command	[this file is not yet included in the current delivery]	
Verify [L2BP 160] by observing that examples of all auxiliary data files mentioned above are present in listings produced by the above two tar commands.		
tar ztvf L2BP_Release1.32A_TDS.tar.gz		
Verify [L2BP 150] by observing that this TDS dataset contains all additional datafiles needed to perform the FAT.	[Note that this FAT depends largely on visual inspection of results, so at the moment it is not clear to me what can be packed into a test script and what not. This needs further discussion after we have agreed on, and completed this list of tests]	
ls *.pdf		
Verify [L2BP 170] by observing that a file named AE- MA- ECMWF-L2BP-001_20061207_SUM_Iss1.2.pdf is		



<i>Procedure / Command</i>	<i>comments/ Observations</i>	<i>passed/ failed</i>
present in the listing generated by the above ls command.		
[similar checks for the other documentation?]		

*Table 4.1.: test procedure for test L2bP\_DEL\_01*

#### 4.1.5 Output Specification

All tar commands produce a listing of files available in the delivered \*.tar.gz packages. The ls command produces a listing of files available in the current directory.

#### 4.1.6 Pass/fail criteria.

The test passes if all mentioned data and documentation files are present. If one or more are missing the test fails.

### 4.2 Test L2bP\_GEN\_01

This test procedure verifies the hardware to be used.

#### 4.2.1 Purpose

Test L2bP\_GEN\_01 verifies requirements: [L2BP 780]

#### 4.2.2 Prerequisites

A test platform must be available.

#### 4.2.3 Input Specification

All commands are entered by the operator via Linux command line.

#### 4.2.4 Output Specification

The commands list some of the properties of the current hardware.

#### 4.2.5 Specific Test Procedure

Commands to be executed are:



<i>Procedure / Command</i>	<i>comments/ Observations</i>	<i>passed/ failed</i>
cat /proc/cpuinfo		
verify that the processor on the current machine is a Pentium 4 or better, with a clock speed of 2.5 GHz. or better.		
cat /proc/meminfo		
verify that the available total RAM memory is 256 MB or more		
df -h		
verify that the disk on which the installation is to be performed has at least 500 MB of free space		
Verify [L2BP 780] by observing that the above 3 conditions are met.		

*Table 4.2.: test procedure for test L2bP\_GEN\_01*

#### 4.2.6 Pass/fail criteria

The test is considered passed if all mentioned hardware properties are equal to or better than the mentioned minimum values. Otherwise the test fails.

### 4.3 Test L2bP\_GEN\_02

This test procedure verifies the formatting and contents of the documentation and the formatting of the source code of the L2bP software.

#### 4.3.1 Purpose

Test L2bP\_GEN\_02 verifies requirements: [GEN 50], [GEN 60], [L2BP 490], [L2BP 530], [L2BP 790], [L2BP 800], [L2BP 810].

#### 4.3.2 Prerequisites

The L2bP software package and documentation must be copied onto the test platform, the package must have been unpacked by using the command "tar zxvf L2BP\_Release1.32A\_source.tar.gz".



#### 4.3.3 Input Specification

All commands are entered by the operator via Linux command line.

#### 4.3.4 Output Specification

tbd





	Document- No. <b>AE- PL- KNMI- L2BP- 001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>20 / 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				

### 4.3.5 Specific Test Procedure

Commands to be executed are:

<i><b>Procedure / Command</b></i>	<i><b>comments/ Observations</b></i>	<i><b>passed/ failed</b></i>
Verify that an up to date copy of the documentation is present in pdf format by giving the command "ls *.pdf".		
Verify [GEN 60] by opening each of these files with a suitable pdf viewer like xpdf, and observing that its contents is readable.		
Verify [GEN 50] by opening each of these files with a suitable pdf viewer like xpdf, and observing that their structure is according to the ECSS standards as tailored by the SOW documents [AD 2] and [AD 3].		
This includes the following files:		
IODD:AE-IF-ECMWF-L2BP-001_*.pdf		
ExtICD: AE-IF-ECMWF-L2BP-002_*.pdf		
SUM: AE-MA-ECMWF-L2BP-001_*.pdf		
SRN: AE-RN-ECMWF-L2BP-001_*.pdf		
SRD: AE-RS-ECMWF-L2B-001_*.pdf		
ATBD: AE_TN_ECMWF_L2P_0023_*.pdf		
EE2BUFRGuide: AE-TN-ECMWF-L2P-0072_*.pdf		
Verify [L2BP 490] by opening the SUM document and observe that in section 4.5 the exceptional manual operation of the L2BP software is described, and in section 7 the additional assimilation interface to be used in the ECMWF IFS context is described.		
Verify [L2BP 530] by opening the ATBD document, selecting some algorithms at random [or all of them if time permits], locate them in the source code, and compare the ATBD description with the source code.		
Verify [L2BP 790] by opening some source code files at random [or all of them if time permits], and observe that they are programmed in one of the permitted languages: Fortran90, c, c++, csh/sh, Python or Make.		
Verify [L2BP 800] by opening the design document [RD 1] and observe that all non- standard Fortran90 calls (which are all collected in the different compiler_features_*.F90 modules in directory support) are listed.	[this is currently not yet the case]	



	Document- No. <b>AE- PL- KNMI- L2BP- 001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>21 / 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				

<i><b>Procedure / Command</b></i>	<i><b>comments/ Observations</b></i>	<i><b>passed/ failed</b></i>
Verify [GEN 40] by opening the design document [RD 1] and observing that the processor consists of a single module (executable)		
Verify [L2BP 810] by starting the matlab package using the command "matlab - nodisplay" and observe that the software version is 7.0.0.0 or newer.		

*Table 4.3.: test procedure for test L2bP\_GEN\_02*

#### **4.3.6 Pass/fail criteria**

The test is considered passed if all requirement tests mentioned in the above Table 4.3 are passed. If one of them fails, this test is considered to have failed.

### **4.4 Test L2bP\_GEN\_03**

This test procedure verifies the installation of the L2bP software.

#### **4.4.1 Purpose**

Test L2bP\_GEN\_03 verifies requirements: [L2BP 20], [GEN 80], [L2BP 80], [L2BP 230]

#### **4.4.2 Prerequisites**

The L2bP software package and documentation must be copied onto the test platform.

#### **4.4.3 Input Specification**

All commands are entered by the operator via Linux command line.



#### **4.4.4 Output Specification**

After this test the software is installed on this platform in working order.

#### **4.4.5 Specific Test Procedure**

Commands to be executed are:



	Document- No. <b>AE-PL-KNMI- L2BP-001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>22/ 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				

<i>Procedure / Command</i>	<i>comments/ Observations</i>	<i>passed/ failed</i>
note the current time		
Install the L2BP software by following the instructions detailed in the SRN document, section 3.		
Verify success of the installation by observing that the Install directory contains an executable named L2B_processor. When executed without JobOrder file by giving the command <code>./install/L2B_processor</code> a help message should be printed, giving instructions how the processor should be used.		
issue the commands: <code>grep -i failed build/Test/testresults.accumulated   wc -l</code> and <code>grep -i passed build/Test/testresults.accumulated   wc -l</code> and verify that no unit test failed, and that 166 unit tests passed.		
note the elapsed time since the start of installation		
Verify [L2BP 20] by observing that a successful installation took less than 4 hours, combined with the observation that all unit tests passed, and that the L2B_processor executable was successfully build.		
When [L2BP 20] is passed, this automatically means that [GEN 80] and [L2BP 80] are passed as well		
Verify [L2BP 230] by observing that in the build directory the files Makeoptions and Systemsettings are ascii files that may be edited by the user if needed, to change configuration options.  (If I understand correctly the settings in the AUX_PAR file are not tested by this requirement)		

*Table 4.4.: test procedure for test L2bP\_GEN\_03*



#### 4.4.6 Pass/fail criteria

The test is considered passed if installation of the L2BP was successful, and was completed within 4 hours.

### 4.5 Test L2bP\_GEN\_04

This test procedure verifies correct handling of mathematical exceptions in the L2bP software.



	Document- No. <b>AE- PL- KNMI- L2BP- 001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>23 / 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				

#### 4.5.1 Purpose

Test L2bP\_GEN\_04 verifies requirements: [GEN 100], [GEN 110], [GEN 120], [GEN\_130]

#### 4.5.2 Prerequisites

The L2bP software must be installed onto the test platform.

#### 4.5.3 Input Specification

All commands are entered by the operator via Linux command line.

#### 4.5.4 Output Specification

t.b.d.



#### 4.5.5 Specific Test Procedure

Commands to be executed are:

<i><b>Procedure / Command</b></i>	<i><b>comments/ Observations</b></i>	<i><b>passed/ failed</b></i>
Verify [GEN 100] by searching the source code (using for example the grep tool) for calls to log(), atan(), '/', etc. and observe that the inputs are properly checked on their valid range (including possible NaN and Inf numbers in the input files) before entering the calculation. Also search the ascii L2BP product and note that no suspicious outputs like NaN or Inf occur. This may be done by using the Write_L2B_Text_Product tool, which should be present in the install directory, on any valid L2B product file.		
Verify [GEN 110] by taking one or more examples found in the above test, and observing in the source code that the variable is set to missing when a non- fatal mathematical exception is found, without producing any log messages. For an example see handling of the log() function in subroutine Calc_Extinction_Iterative() in file opticalproperties.F90 in directory OpticalProperties.	Alternatively some dedicated code might be added to enable triggering such conditions [not yet implemented]	
Verify [GEN 120] and [GEN 130] by taking one or more examples found in the above test, and observing in the source code that the error_flag is set and a proper log message is issued when a fatal mathematical exception is found. For an example see the several checks for negative attenuation numbers in subroutine calc_FP_Molecules_Only() in file opticalproperties.F90 in directory OpticalProperties.	Alternatively some dedicated code might be added to enable triggering such conditions [not yet implemented]	

*Table 4.5.: test procedure for test L2bP\_GEN\_04*



	Document- No. <b>AE- PL- KNMI- L2BP- 001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>24 / 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				

#### 4.5.6 Pass/fail criteria

This test is considered passed if all requirements mentioned in the specific test procedure above in Table 4.5 are passed, and failed otherwise.

### 4.6 Test L2bP\_GEN\_05

This test procedure verifies bit and byte ordering in the L2bP software.

#### 4.6.1 Purpose

Test L2bP\_GEN\_05 verifies requirements: [GEN 10], [GEN 20]

#### 4.6.2 Prerequisites

The L2bP software and associated binary datapack must be installed onto the test platform, and the set of unit tests must have been run at least once (to create the necessary symbolic links in the Test/DirectBinaryIO directory).

#### 4.6.3 Input Specification



All commands are entered by the operator via Linux command line.

#### 4.6.4 Specific Test Procedure

Commands to be executed are:

<i><b>Procedure / Command</b></i>	<i><b>comments/ Observations</b></i>	<i><b>passed/ failed</b></i>
cd \$(SOURCE_DIR)/Test/DirectBinaryIO		
./WriteSomeNumbers		
Verify that the generated file SomeNumbers.bin is written in the native byte order of the current platform. Inspect for example bytes 0E and 0F using the command "hexdump -C SomeNumbers.bin". These bytes should hold the numbers "02 00" for little endian platforms like linux. Example files SomeNumbers.bin.*.expected are provided as reference.		
./TestDirectBinaryIO		
Verify that the file test_file.bin has bigendian byteorder, no matter what platform it runs on. This can be easily checked by using the command "hexdump -C test_file.bin" to verify that for example bytes 14- 17 (hexadecimal 0E- 12) which should contain a 4 byte signed integer (IntAI type) with a value of 4 does indeed contain the bytes "00 00 00 04". Compare this with the output of "hexdump -C SomeNumbers.bin" which writes the IntAul number 2 on a little endian platform like linux as "02 00 00 00" in bytes 38- 41 (hexadecimal		



	Document- No. <b>AE- PL- KNMI- L2BP- 001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>25 / 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				

<i>Procedure / Command</i>	<i>comments/ Observations</i>	<i>passed/ failed</i>
26- 29). This verifies requirements [GEN 10] and [GEN 20]		

*Table 4.6.: test procedure for test L2bP\_GEN\_05*

#### 4.6.5 Output Specification

The WriteSomeNumbers tool produces a binary file named SomeNumbers.bin and writes some information to the screen

The TestDirectBinaryIO tool produces a binary file named test\_file.bin which should always be in bigendian format, no matter what the native format of the platform on which it is installed uses. It also reads this file again, and reads the SomeNumbers.bin file produced by WriteSomeNumbers.

#### 4.6.6 Pass-fail criteria

The generated binary files should be in native and big- endian byteorder as is detailed in the test procedure. If they are not the test fails.

### 4.7 Test L2bP\_HMI\_01

This test procedure verifies the human machine interface for the L2bP software, combined with the general correct functioning of the processor.

#### 4.7.1 Purpose

Test L2bP\_HMI\_01 verifies requirements: [L2BP 30], ~~[L2BP 200]~~, ~~[L2BP 210]~~, [L2BP 250], [L2BP 270], [L2BP 280], [L2BP 290], [L2BP 300], [L2BP 301], [L2BP 310], [L2BP 330], [L2BP 381], [L2BP 382] [L2BP 440], [L2BP 500], [L2BP 510], [L2BP 520], ~~[L2B 690]~~.

#### 4.7.2 Prerequisites

The L2bP software must be installed onto the test platform.

Additional L1B product files produced by the L1BP are available for testing.

An example JobOrder file produced by the real ThinLayer (not the Emulator!) is available.

#### 4.7.3 Input Specification

All commands are entered by the operator via Linux command line.



#### 4.7.4 Output Specification

t.b.d.

#### 4.7.5 Specific Test Procedure



Commands to be executed are:



	Document- No. <b>AE- PL- KNMI- L2BP- 001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>26 / 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				

<i>Procedure / Command</i>	<i>comments/ Observations</i>	<i>passed/ failed</i>
cd \$(BUILD)/Test/main		
./L2B_processor JobOrder.test12.xml <u>-LogFile Test.log &gt; textoutput.txt</u>		
Verify [L2BP 30], <del>[L2BP 200]</del> , [L2BP 440] by observing that the above commands produced a valid L2B product file named AE_TEST_ALD_U_T_2B_*.DBL/HDR.		
<u>Verify [L2BP 381] by observing that all log messages have been written to the logfile named Test.log</u>		
<u>Verify [L2B 690] by observing that all log messages available in the file Test.log are also available in the text output which was redirected to the file textoutput.txt. This is how a log file would be generated when running in the ThinLayer environment (i.e. by redirecting the standard output).</u>		
Copy a JobOrder file produced by the real ThinLayer to the current directory (assume it is named JobOrder.ThinLayer.xml)		
./L2B_processor JobOrder.ThinLayer.xml		
Verify [L2BP 310] by observing that the above commands produced a valid L2B product file named AE_TEST_ALD_U_T_2B_*.DBL/HDR		
<del>Verify [L2BP 210] by observing that the L2BP can be used immediately after a previous run ended.</del>		
cp JobOrder.test12.xml JobOrder.FAT_test.xml		
Edit JobOrder.FAT_test.xml and point it to any valid L1B file that needs to be tested.		
./L2B_processor JobOrder.FAT_test.xml > textoutput.txt		
Verify [L2BP 250] and [L2BP 520] by observing that the above commands produced a valid L2B product file named AE_TEST_ALD_U_T_2B_*.DBL/HDR		
Verify [L2BP 270] by observing in the textoutput.txt file that all Aux_Par file settings present in the input file AE_TEST_AUX_PAR_2B_20050112T000000_20110112T000000_0012.EEF are printed on lines starting with "L2BP_Settings%L2B_AuxPar". If desired edit this file, run the processor again, and observe that the changed entries are imported in the software.		
Verify [L2BP 280] by observing that the Aux_Clim file named	[not yet implemented]	



	Document- No. <b>AE- PL- KNMI- L2BP- 001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>27 / 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				

<i>Procedure / Command</i>	<i>comments/ Observations</i>	<i>passed/ failed</i>
AE_TEST_AUX_CLM_AX_20050331T000000_20111231T00000_0000.DBL/HDR is read.		
Verify [L2BP 290] by observing that the Aux_RBC file named AE_TEST_AUX_RBC_AX_20060703T123456_20070703T123456_0001 is read. (search for the string Load_Rayl_Brill_Tables in the textoutput.txt file)		
Verify [L2BP 300] by observing that the Aux_Met file named AE_TEST_AUX_MET_12_20041201T000000_20041201T003000_0001 is read. (search for the string "reading aux met data" in the textoutput.txt file)		
Verify [L2BP 301] by observing that the Aux_Cal file named [t.b.d.] is read.	[not yet implemented]	
Verify [L2BP 330] by observing that a proper exit code, as defined in the L2b External ICD [AD 9], is produced when the processor ends.	[not yet implemented]	
Verify [L2BP 500] by observing that all complete observations in the input L1B product file have been processed and written to the L2B product file.  Check the number of available observations (BRC's) in the L1B input product by checking the Total_Num_of_Observations field of the SPH in the L1B product headerfile.  Check the number of processed observations (BRC's) in the L2B product by searching for the string "nr of processed BRC" in the file extoutput.txt file. Also check the Num_BRC field of the SPH in the L2B product headerfile.		
Edit the file JobOrder.FAT_test.xml to limit the timerange of the L1B products to be processed, and run the processor again:		
./L2B_processor JobOrder.FAT_test.xml > textoutput.txt		
Verify [L2BP 510] by observing that all observations in the given timerange have been processed.	<del>note that the timerange selection is not yet implemented in the L2BP. The L2BP will always process all observations in the L1BP, which is not in violation of the formulation of this requirement. [i.e. there is no</del>	



<i>Procedure / Command</i>	<i>comments/ Observations</i>	<i>passed/ failed</i>
	<del>requirement to not process observations outside the given timerange</del>	
Edit the file JobOrder.FAT_test.xml to change the loglevel of the stdout channel, and run the processor again:		
./L2B_processor JobOrder.FAT_test.xml > textoutput.txt		
Verify [L2BP 382] by observing that the amount of log messages has changed according to the choosen log level.		

*Table 4.7.: test procedure for test L2bP\_HMI\_01*

#### 4.7.6 Pass/fail criteria

This test is considered passed if all requirements mentioned in the specific test procedure above in Table 4.7 are passed, and failed otherwise.

### 4.8 Test L2bP\_EXT\_01

This test procedure verifies the external interface for the L2bP software (especially the ThinLayer emulator interface).

#### 4.8.1 Purpose

Test L2bP\_EXT\_01 verifies requirements: [GEN 90 ], [L2BP 470], [L2BP 383], [L2BP 460], [L2BP 480]

#### 4.8.2 Prerequisites

The L2bP software must be installed onto the test platform.

The ThinLayer Emulator must be installed onto the test platform.

#### 4.8.3 Input Specification

All commands are entered by the operator via Linux command line, or via the ThinLayer emulator user interface.

#### 4.8.4 Output Specification

t.b.d.

#### 4.8.5 Specific Test Procedure

Commands to be executed are:



<i>Procedure / Command</i>	<i>comments/ Observations</i>	<i>passed/ failed</i>
Insert the L2BP in the ThinLayer emulator (TLE) by following the commands detailed in [AD 9].	[note that this description of this procedure is not yet available in this document)	
Start the TLE user interface: /usr/acs/bin/TISandaloneMMI		
select the example order nr. 5 supplied by the L2BP delivery, and run the processor by clicking the "process" button.		
Note that error and/or warning messages appear in the log and control panel of the user interface of the TLE. This verifies [L2BP 383]		
Check that both the L2BP processor and the ReportGenerator task finished successfully. This verifies [GEN 90], [L2BP 470], [L2BP 480]		
Verify [L2BP 460] by observing that the above test run generated a product report.		
Verify [L2BP 335] by observing that the above test run generated a Product List file, as defined in the L2B External ICD [AD 9]		

*Table 4.8.: test procedure for test L2bP\_EXT\_01*

#### 4.8.6 Pass/fail criteria

This test is considered passed if all requirements mentioned in the specific test procedure above in Table 4.8 are passed, and failed otherwise.

### 4.9 Test L2bP\_EXT\_02

This test procedure verifies the formatting of the example files needed to interface the L2bP software with the ThinLayer emulator.

#### 4.9.1 Purpose

Test L2bP\_EXT\_02 verifies requirements: [L2BP 100], [L2BP 320], [L2BP 390], [L2BP 400], [L2BP 410], [L2BP 420], [L2BP 430].

#### 4.9.2 Prerequisites

The L2bP software must be installed onto the test platform...



### 4.9.3 Input Specification

All commands are entered by the operator via Linux command line.

### 4.9.4 Output Specification

- first expected test result should be ...
- second expected testresult should be ...

### 4.9.5 Specific Test Procedure

Commands to be executed are:



<i><b>Procedure / Command</b></i>	<i><b>comments/ Observations</b></i>	<i><b>passed/ failed</b></i>
Verify [L2BP 100] by reviewing chapter 4.2.1 of [AD 9]		
Verify [L2BP 320] by comparing the example JobOrder file JobOrder.test12.xml in directory \$(BUILD)/Test/main with chapter 4.2.1 of [AD 9]		
Verify [L2BP 390] by comparing the example Workstation Configuration file named WorkstationConfigurationFile.xml in directory \$(BUILD)/ThinLayer with chapter 4.1.1 of [AD 9]		
Verify [L2BP 400] by observing that the example Workstation Configuration file named WorkstationConfigurationFile.xml in directory \$(BUILD)/ThinLayer lists the allowed order types of the L2BP		
Verify [L2BP 410] by comparing the example Task table file named TaskTable.AE_L1B_L2B_WIND.xml in directory \$(BUILD)/ThinLayer with chapter 4.1.2 of [AD 9]		
Verify [L2BP 420] by observing that the example Task table file named TaskTable.AE_L1B_L2B_WIND.xml in directory \$(BUILD)/ThinLayer lists two tasks (each in its own pool), one for the actual L2B_Processor, and one for the ReportGenerator.		
Verify [L2BP 430] by observing that the example Task table file named TaskTable.AE_L1B_L2B_WIND.xml in directory \$(BUILD)/ThinLayer lists 5 input files and 1 output file for the L2B_processor, and 1 input file and 1 output file for the ReportGenerator task.		

*Table 4.9.: test procedure for test L2bP\_EXT\_02*

### 4.9.6 Pass/fail criteria

This test is considered passed if all requirements mentioned in the specific test procedure in Table 4.9 above are passed, and failed otherwise.



	Document- No. <b>AE-PL-KNMI- L2BP-001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>31/ 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				

## 4.10 Test L2bP\_EXT\_03

This test procedure verifies the ability of the L2BP software to issue the different types of log messages on the stdout and stderr channels. It also verifies the respons of the L2BP to external signals.

### 4.10.1 Purpose

Test L2bP\_EXT\_03 verifies requirements: [GEN 30], [L2BP 340], [L2BP 350], [L2BP 360], [L2BP 370], [L2BP 380], [L2BP 450], [L2BP 670]

### 4.10.2 Prerequisites

The L2bP software must be installed onto the test platform...

### 4.10.3 Input Specification

All commands are entered by the operator via Linux command line.

### 4.10.4 Output Specification



t.b.d.

### 4.10.5 Specific Test Procedure

Commands to be executed are:

<i><b>Procedure / Command</b></i>	<i><b>comments/ Observations</b></i>	<i><b>passed/ failed</b></i>
With a text editor open the file textoutput.txt created during the L2bP_HMI_01 tests described in section 4.7		
Verify [GEN 30] by observing that this file contains proper ASCII text, which proves the proper bit and byte ordering is used on the stdout and stderr interfaces as well		
Verify [L2BP 340] by searching for the string "[I]" and observing that some examples have been printed to stdout		
Verify [L2BP 350] by searching for the string "[P]" and observing that some examples have been printed to stdout		
Verify [L2BP 360] by searching for the string "[W]" and observing that some examples have been printed to stdout		
Verify [L2BP 370] by searching for the string "[E]" and observing that some examples have been printed to stdout.	[only if an error is provoked] [Alternatively some	



	Document- No. <b>AE- PL- KNMI- L2BP- 001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>32/ 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				

<i>Procedure / Command</i>	<i>comments/ Observations</i>	<i>passed/ failed</i>
[the order type Backlog_Processing in the JobOrder file generates an error at the moment, which can be used for this example, even though the processor ignores it]	dedicated code might be added to enable triggering such conditions [not yet implemented]	
Verify [L2BP 380] by searching for the string "[D]" and observing that some examples have been printed to stdout	<del>[not in use yet]</del>	
close the text editor		
Rerun the command: ./L2B_processor JobOrder.test12.xml in directory: \$(BUILD)/Test/main as specified in Table 4.7		
Verify [L2BP 450] by giving a kill or terminate signal (type control- c) when the above command is running, and observing that the processing is aborted. Note that the verification of [L2BP 310] in Table 4.7 already showed that under normal circumstances the processor does not need such a signal to stop after it has been launched.		
cp JobOrder.test12.xml JobOrder.testFAT.xml		
edit JobOrder.testFAT.xml and change the Stderr_Log_Level field to any other valid log level except NOOP. Then run the following command		
./L2B_processor JobOrder.testFAT.xml > text_stdout.txt		
Verify [L2BP 670] by observing that log messages are printed to the screen now (which is only connected to stderr in this test case), since all output to stdout is redirected to the file text_stdout.txt		

*Table 4.10.: test procedure for test L2bP\_EXT\_03*



#### 4.10.6 Pass/fail criteria

This test is considered passed if all requirements mentioned in the specific test procedure in Table 4.10 above are passed, and failed otherwise.

#### 4.11 Test L2bP\_PROD\_01

This test procedure verifies the ability of the L2BP software, after a successful run, to generate a product report according to the specifications, and verifies that it (only) produces the messages specified in the L2BP external ICD [AD 9]



	Document- No.	Issue:	Date:	Page:	
	AE- PL- KNMI- L2BP- 001	V 1.32	11.04.2007	33 / 62	
	Doc.- Title:				
	Aeolus Level 2B Processor Software Verification & Validation Plan				

#### 4.11.1 Purpose

Test L2bP\_PROD\_01 verifies requirements: [L2BP 580], [L2BP 590], [L2BP 600], [L2BP 610], [L2BP 620], [L2BP 630], [L2BP 640], [L2BP 680]

Not tested yet, but will be tested here once it becomes clear what we have to do here:

[L2BP 650]

#### 4.11.2 Prerequisites

The L2bP software must be installed onto the test platform.

Test L2bP\_HMI\_01 defined in section 4.7 must have been run before starting this test.

#### 4.11.3 Input Specification

All commands are entered by the operator via Linux command line.



#### 4.11.4 Output Specification

t.b.d.

#### 4.11.5 Specific Test Procedure

Commands to be executed are:



	Document- No. <b>AE- PL- KNMI- L2BP- 001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>34 / 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				



<i>Procedure / Command</i>	<i>comments/ Observations</i>	<i>passed/ failed</i>
produce a product report from the file textoutput.txt created during the L2bP_HMI_01 tests described in section 4.7, using the following commands:		
cd \$(BUILD)/Test/main		
../../Scripts/L2B_ReportGenerator.py -l textoutput.txt		
open the product report file, which has the default name DefaultProductReport, with a text editor.		
Verify [L2BP 580] by observing that a Processor_Version field is present		
Verify [L2BP 590] by searching for the string " <del>...loaded</del> " and observing that the input files are listed.	<del>{this is not yet implemented}</del>	
Verify [L2BP 600] by searching for the string " <del>...written productfile</del> " and observing that all output products are listed	<del>{this is not yet implemented}</del>	
Verify [L2BP 610] by searching for the string " <del>...size of binary productfile</del> " and observing that the sizes of all output products are reported	<del>{this is not yet implemented}</del>	
Verify [L2BP 620] by searching for the string "..." and observing that an exit code indicating success or failure field is present	[this is not yet implemented]	
Verify [L2BP 630] by searching for the string ">[W<]" or ">[E<]" and observing that warnings (and errors if applicable) are reported.		
Verify [L2BP 640] by searching for the strings "Start <del>deing</del> L2B-Processor" and "Finished L2B-Processor", and observing that these messages have a date-time tag	<del>{note that this first msg is not yet implemented. As a replacement the first msg of any type, reported in the report, may be used}</del>	
Verify [L2BP 650] by observing that ... [tbd]		
Verify [L2BP 680] by that all informational/Warning/Error messages issued by the processor and written to the ProductReport are listed in the L2BP external ICD [AD 9]	[this list needs to be added to the ICD]	

*Table 4.11.: test procedure for test L2bP\_PROD\_01*

#### 4.11.6 Pass/fail criteria

This test is considered passed if all requirements mentioned in the specific test procedure



	Document- No. <b>AE- PL- KNMI- L2BP- 001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>35 / 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				

defined in Table 4.11 above are passed, and failed otherwise.

## 4.12 Test L2bP\_PROD\_02

This test procedure verifies the ability of the L2BP software to react properly on missing or corrupted input dat, and missing input files, and verifies a proper warning/error messages is issued.

### 4.12.1 Purpose

Test L2bP\_PROD\_02 verifies requirements: [L2BP 540], [L2BP 550], [L2BP 560], [L2BP 570], [L2BP 660]

### 4.12.2 Prerequisites

The L2bP software must be installed onto the test platform.

The TDS dataset must be unpacked in directory \$(TDS)

### 4.12.3 Input Specification

All commands are entered by the operator via Linux command line.

### 4.12.4 Output Specification

t.b.d.

### 4.12.5 Specific Test Procedure

Commands to be executed are:

<i><b>Procedure / Command</b></i>	<i><b>comments/ Observations</b></i>	<i><b>passed/ failed</b></i>
if not yet available, unpack the TDS dataset with the following command:		
tar ztvf L2BP_Release1.32A_TDS.tar.gz		
cd \$(TDS)/extra/Data/...	[not yet available]	
\$(INSTALL)/L2B_processor JobOrder.L1B_Corrupt.xml	[not yet available]	
Or alternatively (as long as we have not produced a suitable test L1B product containing invalid or missing data):		
use a hexadecimal editor to edit the L1B input product used for the previously used test L2bP_HMI_01 in section 4.7, change some numbers in it.		
cd \$(BUILD)/Test/main		
hexedit [any inputfile]		



<i>Procedure / Command</i>	<i>comments/ Observations</i>	<i>passed/ failed</i>
./L2B_processor JobOrder.test12.xml		
After running the L2BP on a test file containing some missing or corrupt L1B inputs using the command above, verify [L2BP 540] by observing that the problematic inputs are detected, flagged and reported, while the remaining correct inputs are processed anyway.	<u>[to be expanded to test all conditions mentioned in Table 4.22]</u>	
Remove one of the necessary input files by renaming it, and run the test again:		
mv AE_TEST_AUX_PAR_2B_20050112T000000_20110112T000000_0012.EEF AE_TEST_AUX_PAR_2B_20050112T000000_20110112T000000_0012.EEF.orig		
./L2B_processor JobOrder.test12.xml		
Verify [L2BP 550] by observing that the L2BP stops and issues a proper error message about this missing file.		
Verify [L2BP 560] by observing that a suitable error is written to the event log, about the missing inputfile.		
Verify [L2BP 570] by observing that the last successfully completed processing step is reported when the L2BP fails irrecoverably, and by observing that no more steps are reported, except for the error that occurred.		
Verify [L2BP 660] by observing that all above produced warnings and errors have been logged.		

*Table 4.12.: test procedure for test L2bP\_PROD\_02*

#### 4.12.6 Pass/fail criteria

This test is considered passed if all requirements mentioned in the specific test procedure above in Table 4.12 are passed, and failed otherwise.

### 4.13 Test L2bP\_PROD\_03

This test procedure verifies the ability of the L2BP software to process a very large L1B input product, holding 8 orbits of data.

#### 4.13.1 Purpose

Test L2bP\_PROD\_03 verifies requirement: [L2BP 302]



#### 4.13.2 Prerequisites

The L2bP software must be installed onto the test platform.

The additional test dataset must be unpacked in directory \$(TDS)

#### 4.13.3 Input Specification

All commands are entered by the operator via Linux command line.

#### 4.13.4 Output Specification

t.b.d.

#### 4.13.5 Specific Test Procedure

Commands to be executed are:

<i>Procedure / Command</i>	<i>comments/ Observations</i>	<i>passed/ failed</i>
cd \$(TDS)/extra/Data/...	[not yet available]	
\$(INSTALL)/L2B_processor JobOrder.L1B_8orbits.xml	[not yet available]	
Verify [L2BP 302] by observing that the above command successfully processed an L1B product holding 8 orbits of data		

*Table 4.13.: test procedure for test L2bP\_PROD\_03*

#### 4.13.6 Pass/fail criteria

This test is considered passed if all requirements mentioned in the specific test procedure above in Table 4.13 are passed, and failed otherwise.

### 4.14 Test L2bP\_PROD\_04

This test procedure verifies the correct format of the L2B output product.

#### 4.14.1 Purpose

Tests after a successful run (so following PROD\_02)

Test L2bP\_PROD\_04 verifies requirements: [L2BP 110], [L2BP 162], [L2BP 260], [L2BP 700], [L2BP 710], [L2BP 720], [L2BP 730], [L2BP 740], [L2BP 760]

#### 4.14.2 Prerequisites

The L2bP software must be installed onto the test platform.

Test L2bP\_HMI\_01 defined in section 4.7 must have been run



#### 4.14.3 Input Specification

All commands are entered by the operator via Linux command line.

#### 4.14.4 Output Specification

t.b.d.



#### 4.14.5 Specific Test Procedure

Commands to be executed are:

<i><b>Procedure / Command</b></i>	<i><b>comments/ Observations</b></i>	<i><b>passed/ failed</b></i>
cd \$(BUILD)/Test/main		
../L2BC_file_handling/Write_L2B_Text_Product \$(L2B_Product)		
Verify [L2BP 110] and [L2BP 162] by observing that the above command converted the L2B product generated by test L2bP_HMI_01 in section 4.7 to plain ascii.		
Open the *.HDR and *.DBL files of this L2B product in a text editor. Also open the *.DBL file of this L2B product using a hexdump utility.		
Verify [L2BP 260] and [L2BP 700] (these requirements are identical i.m.o.) by observing that the format of the just opened files is according to the L2B IODD [AD 8]		
Verify [L2BP 710] by observing that the L2B KVT section has a field which gives the JobOrder type that was used.	[not yet implemented]	
Verify [L2BP 720] by observing that the L2B KVT section has a fields giving quality statistics of the generated product, as specified in the L2B IODD [AD 8].		
Verify [L2BP 730] by observing that a Main Product Header is present in the L2B product, and has a structure as specified in the L2B IODD [AD 8].		
Verify [L2BP 740] by observing that a Specific Product Header is present in the L2B product, and has a structure as specified in the L2B IODD [AD 8].		
Verify [L2BP 760] by observing that the Specific Product Header in the L2B product, contains several data set descriptors, which have a structure as specified in the L2B IODD [AD 8].		

*Table 4.14.: test procedure for test L2bP\_PROD\_04*



	Document- No. <b>AE-PL-KNMI- L2BP-001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>39/ 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				

#### 4.14.6 Pass/fail criteria

This test is considered passed if all requirements mentioned in the specific test procedure above in Table 4.14 are passed, and failed otherwise.

### 4.15 Test L2bP\_PERF\_01

This test procedure verifies the ability of the L2BP software to process a L1B input product, holding a full orbit of data, within 30 minutes, on a specific target platform.

#### 4.15.1 Purpose

Test L2bP\_PERF\_01 verifies requirements: [L2BP 770]

#### 4.15.2 Prerequisites

The hardware of the testplatform must have been verified using test [L2bP\_GEN\_01] described in section 4.2

The L2bP software must be installed onto the test platform.

The additional test dataset must be unpacked in directory \$(TDS)

#### 4.15.3 Input Specification

All commands are entered by the operator via Linux command line.

#### 4.15.4 Output Specification

t.b.d.

#### 4.15.5 Specific Test Procedure

Commands to be executed are:



<i>Procedure / Command</i>	<i>comments/ Observations</i>	<i>passed/ failed</i>
cd \$(TDS)/extra/Data/...	[not yet available]	
note the current time		
\$(INSTALL)/L2B_processor JobOrder.L1B_fullorbit.xml	[not yet available]	
note the elapsed time since the start of processor		
Verify [L2BP 770] by observing that the above command successfully processed an L1B product holding a full orbit of data within 30 minutes		

*Table 4.15.: test procedure for test L2bP\_PERF\_01*

#### 4.15.6 Pass/fail criteria

This test is considered passed if all requirements mentioned in the specific test procedure



	Document- No. <b>AE- PL- KNMI- L2BP- 001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>40 / 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				

above in Table 4.15 are passed, and failed otherwise.

## 4.16 Test L2bP\_TOOLS\_01

This test procedure verifies the presence of tools to inspect/view the auxiliary input datafiles used by the L2BP software.

### 4.16.1 Purpose

Test L2bP\_TOOLS\_01 verifies requirement: [L2BP 90]

### 4.16.2 Prerequisites

The L2bP software must be installed onto the test platform.

### 4.16.3 Input Specification

All commands are entered by the operator via Linux command line.

### 4.16.4 Output Specification

t.b.d.

### 4.16.5 Specific Test Procedure

Commands to be executed are:

<i><b>Procedure / Command</b></i>	<i><b>comments/ Observations</b></i>	<i><b>passed/ failed</b></i>
cd \$(BUILD)/Test/RBC_FileHandling		
TestReadRBCdata AE_TEST_AUX_RBC_AX_20060703T123456_20070703T 123456_0001		
cd \$(BUILD)/Test/L2B_AuxPar_file_handling		
Test_Read_L2B_AuxPar_file AE_TEST_AUX_PAR_2B_20050112T000000_20110112T 000000_0012.EEF		
cd \$(BUILD)/Test/AMD_file_handling		
TestReadAMDdata AE_TEST_AUX_MET_12_20041201T000000_20041201T 003000_0001		
cd \$(BUILD)/Test/AuxClim_file_handling		
TestReadAuxClimData AE_TEST_AUX_CLM_AX_20010312T123456_20110312T 123456_0001	[not yet implemented]	



<i>Procedure / Command</i>	<i>comments/ Observations</i>	<i>passed/ failed</i>
Verify [L2BP 90] by observing that each of the above used tools converts one of the auxiliary input files into plain ascii, and that such a tool is available for all 4 defined auxiliary filetypes.		

*Table 4.16.: test procedure for test L2bP\_TOOLS\_01*

#### **4.16.6 Pass/fail criteria**

This test is considered passed if all requirements mentioned in the specific test procedure above in Table 4.16 are passed, and failed otherwise.

### **4.17 Test L2bP\_TOOLS\_02**

This test procedure verifies the presence of tools to display graphically the auxiliary input datafiles used by the L2BP software (at least the Aux.Cal file).

(and maybe also the L1B input product???)

#### **4.17.1 Purpose**

Test L2bP\_TOOLS\_02 verifies requirements: [L2BP 130]

#### **4.17.2 Prerequisites**

The L2bP software must be installed onto the test platform.

#### **4.17.3 Input Specification**

All commands are entered by the operator via Linux command line.

#### **4.17.4 Output Specification**

t.b.d.

#### **4.17.5 Specific Test Procedure**

Commands to be executed are:



<i>Procedure / Command</i>	<i>comments/ Observations</i>	<i>passed/ failed</i>
cd \$(build)/Matlab_Tools/TOOLS		
run the tool to display the auxiliary calibration data graphically <sup>1</sup>	(see footnote)	
Verify [L2BP 130] by observing that the above command produced some graphics holding displaying this auxiliary calibration data.		

*Table 4.17.: test procedure for test L2bP\_TOOLS\_02*

#### 4.17.6 Pass/fail criteria

This test is considered passed if all requirements mentioned in the specific test procedure above in Table 4.17 are passed, and failed otherwise.

### 4.18 Test L2bP\_TOOLS\_03

This test procedure verifies the presence of tools to generate the auxiliary input datafiles used by the L2BP software.

#### 4.18.1 Purpose

Test L2bP\_TOOLS\_03 verifies requirement: [L2BP 161]

#### 4.18.2 Prerequisites

The L2bP software must be installed onto the test platform.

#### 4.18.3 Input Specification

All commands are entered by the operator via Linux command line.

#### 4.18.4 Output Specification

t.b.d.

#### 4.18.5 Specific Test Procedure

Commands to be executed are:

<i>Procedure / Command</i>	<i>comments/ Observations</i>	<i>passed/ failed</i>
cd \$(build)/Test/main		

<sup>1</sup> Remark that display using a text or xml editor would do. Therefore this test may probably be combined with [L2BP 90] which is tested in L2bP\_TOOLS\_01. (see email H. Nett, 31-jan-2007)



<i>Procedure / Command</i>	<i>comments/ Observations</i>	<i>passed/ failed</i>
using a text editor open AE_TEST_AUX_PAR_2B_20050101T000000_20110101T000000_0001.EEF		
Observe that this file may be edited using a simple text editor		
cd \$(build)/Test/AuxClim_file_handling		
./TestWriteAuxClimData		
Observe that this above command has created a new AUX_CLM product named: AE_TEST_AUX_CLM_AX_20010312T123456_20110312T123456_0001		
cd \$(build)/Test/AMD_file_handling		
./TestWriteAMDdata		
Observe that this above command has created a new AUX_MET product named: AE_TEST_AUX_MET_12_20041201T000000_20041201T003000_0001		
cd \$(build)/Test/RBC_FileHandling		
./GenerateRBCdata - RBCModel Tenti		
Observe that this above command has created a new AUX_RBC product named: AE_TEST_AUX_RBC_AX_20060703T123456_20070703T123456_0001		
Verify [L2BP 161] by observing that 3 of the 4 defined auxiliary files have been generated by the above commands, and the 4th can easily be copied and edited by a simple text editor.		

*Table 4.18.: test procedure for test L2bP\_TOOLS\_03*

#### 4.18.6 Pass/fail criteria

This test is considered passed if all requirements mentioned in the specific test procedure above in Table 4.18 are passed, and failed otherwise.

#### 4.19 Test L2bP\_TOOLS\_04

This test procedure verifies the presence of a EE2BUFR conversion tool for L1B product files.



#### 4.19.1 Purpose

Test L2bP\_TOOLS\_04 verifies requirements ~~(on the EE2BUFR tool):~~

~~{no requirements defined yet in the SRD}:~~

~~Run the EE2BUFR tool on any L1B file, then check the contents of the generated BUFR file by reading it and producing an ascii dump of the result. [L1B- EE2BUFR 10], [L1B- EE2BUFR 20], [L1B- EE2BUFR 30], [L1B- EE2BUFR 80], [L1B- EE2BUFR 90], [L1B- EE2BUFR 110], [L1B- EE2BUFR 120], [L1B- EE2BUFR 170], [L1B- EE2BUFR 190].~~

#### 4.19.2 Prerequisites

The L2bP software package and documentation must be copied ~~installed on~~ onto the test platform.

~~The BUFR tool must be installed onto the test platform.~~

#### 4.19.3 Input Specification

All commands are entered by the operator via Linux command line.

#### 4.19.4 Output Specification

t.b.d.

#### 4.19.5 Specific Test Procedure

Commands to be executed are:



<i>Procedure / Command</i>	<i>comments/ Observations</i>	<i>passed/ failed</i>
<u>Give the command:</u> <u>tar tvfz L2BP_Release1.32A_source.tar.gz   grep</u> <u>BUFR_file_handling</u> <u>and verify [L1B- EE2BUFR 10] by observing that a file</u> <u>named L1B_ee2bufr.F90 has been delivered, as well as</u> <u>the needed modules and makefiles to compile it.</u>		
<u>t.b.d. Install the L1B EE2BUFR tool independantly from</u> <u>any previous L2BP installation by following the</u> <u>instructions detailed in [AD 11], or alternatively issue</u> <u>the command sequence below:</u>		
<u>give the commands:</u> <u>cd \$(build)</u> <u>source Scripts/use_g95 (or any other compiler desired)</u> <u>./Set Makeoptions</u> <u>make -f Makefile.aeolus ee2bufr</u> <u>make -f Makefile.aeolus ee2bufrtest</u>		
<u>Verify [L1B- EE2BUFR 20] by observing that the above</u> <u>command sequence has compiled and tested only a selection</u>		





<i>Procedure / Command</i>	<i>comments/ Observations</i>	<i>passed/ failed</i>
<u>of the L2BP software, and has produced the tool named L1B_ee2bufr</u>		
<u>give the commands:</u> <u>cd \$(build)</u> <u>cd Test/BUFR file handling</u> <u>setenv BUFR TABLES \$(build)/BUFR tables/</u> <u>TestL1B_ee2bufr</u> <u>AE TEST ALD U N 1B 20071002T000007536 000028</u> <u>000 000004 0001 1 2</u>		
<u>Verify [L1B-EE2BUFR30] by observing that the command sequence given above has decoded the specified L1B file in BUFR format to a readable ASCII format.</u>		
<u>give the command:</u> <u>L1B_ee2bufr</u> <u>AE TEST ALD U N 1B 20071002T000007536 000028</u> <u>000 000004 0001</u>		
<u>Verify [L1B-EE2BUFR80] by observing that the command sequence given above has accepted a valid L1B product file as input, and has produced a BUFR formatted L1B product file. Verify [L1B-EE2BUFR110] by observing that the tool was used from the Unix commandline.</u>		
<u>Verify [L1B-EE2BUFR120] by observing the exit code issued by the L1B_ee2bufr tool used above.</u>		
<u>Verify [L1B-EE2BUFR170] by observing that in the above example all complete observations in the converted L1B product file have been written to the L1B BUFR file.</u>		
<u>give the commands:</u> <u>cd \$(build)/BUFR_install/bufr_000310/examples</u> <u>make ARCH=linux CNAME= compiler decode bufr</u> <u>cd Test/BUFR file handling</u> <u>\$(build)/BUFR_install/bufr_000310/examples</u> <u>\$(build)/BUFR_install/bufr_000310/examples/decode_b</u> <u>ufr -i</u> <u>AE TEST ALD U N 1B 20071002T000001986 000028</u> <u>000 000004 0002.BUFR</u> <u>[answer the given questions with: y, n, n, 1, [ret], [ret],</u> <u>[ret], [ret], y, 1, 1]</u>		
<u>Verify [L1B-EE2BUFR90] by observing that the independant decode_bufr tool provided by ECMWF, used in the command sequence above, demonstrated that the L1B BUFR-product was formatted as specified in [AD 10].</u>		



	Document- No. <b>AE-PL-KNMI- L2BP-001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>46 / 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				

<i>Procedure / Command</i>	<i>comments/ Observations</i>	<i>passed/ failed</i>
<u>Verify [L1B-EE2BUFR190] by feeding a L1B product file that violates the L1B IODD [RD 2], and observing that the tool detects the problems in this file.</u>		

*Table 4.19.: test procedure for test L2bP\_TOOLS\_04*

#### **4.19.6 Pass/fail criteria**

This test is considered passed if all requirements mentioned in the specific test procedure above in Table 4.19 are passed, and failed otherwise.



## A Verification Cross- Reference Matrix

A verification- cross- reference- matrix (VCRM) provides a mapping between the L2bP requirements as listed in [AD 7], the software components that satisfy them, and the test cases that will be used to verify them.

The L2bP VCRM has several columns: Software Requirement, Test Case Name, Verification Method, and Software Component/remark. Their meaning and possible entries in the matrix are described in detail below.

- **Software Requirement:** This column provides the number and text of all the requirements specified in the Aeolus L2bP and End- To- End Simulator Software Requirements Specification, Document [AD 7]
- **Test Case Name:** This column cross- references the Software Requirement in the first column with the test case that will be used to show that this requirement is satisfied. The identifiers used in the matrix for these test cases are summarised in Table 3.2.
- **Verification Method:** This column specifies the verification method that will be used to verify that the Test Procedure in the previous column satisfies the Software Requirement of the first column. Table 4.20 describes the different verification methods.

Verification Method	Verification Method Identifier	Description
Test	T	This method involves verification by executing a test in accordance with an approved test procedure. This type of test includes <b>Functional Tests</b> that verify the capability of the component to perform that function and <b>Quantitative Tests</b> that measure a quantitative performance characteristic. The test involves executing the test procedures and observing the desired results without quantitative assessment of the result.
Analysis	A	This is the analytical determination that a requirement is met. Verification by analysis will constitute one or more of the following: engineering evaluation, mathematical modelling and technical or mathematical deduction. This analysis depends upon test results, published specifications, as well as design considerations and operating conditions.
Review and Inspection	R & I	Verification by review or inspection will constitute a visual examination of the item, together with a review of the descriptive documentation and/or a comparison of the physical characteristics with those identified in the applicable specification.



Verification Method	Verification Method Identifier	Description
Demonstration	D	This method is similar to a Functional Test but is informal in nature and may not have specific input data sets or procedures.



**Table 4.20 L2bP Verification Methods**

- **Software Component/Remark:** This column cross- references the Software Requirement in the first column with the first level L2bP Software Components, described in Level 2 Design Document. Sometimes this column is used to add additional remarks on the test case.

[Note that I rephrased some of the requirements, this is indicated in **yellow**. Please check if you can agree on these modifications and if so change them in the requirements document accordingly. Also some additional remarks, todos and questions from me are marked in **yellow**.]



	<i>requirement</i>	<i>Test Case Name</i>	<i>Verifi- cation Method</i>	<i>Software Component/remark</i>
[GEN 10]	The L2BP shall use bit 0 as the most significant bit for each external interface ( <del>=file</del> )	L2bP_GEN_05	D	all file reading and writing code in the DirectBinaryIO module
[GEN 20]	The L2BP shall use byte 0 as the most significant byte for each external interface ( <del>=file</del> )	L2bP_GEN_05	D	dito
[GEN 30]	The L2BP shall be designed to transmit bit 0 of byte 0 first over each external interface	L2bP_EXT_03	A	
[GEN 40]	The L2BP software system shall be designed to separate process tasks into modular configuration items	L2bP_GEN_02	A	
[GEN 50]	All documentation produced in the context of this contract shall be produced in accordance with the pertinent European Cooperation for Space Standardisation (ECSS) standards, as tailored by the Statement of Work documents [AD 2], and [AD 3].	L2bP_GEN_02	R & I	-
[GEN 60]	All documentation produced in the context of this contract shall be delivered in the following formats: PDF (drafts and final versions) and RTF (final versions only). <u>MS Word shall be used for intermediate versions and drafts, for the</u>	L2bP_GEN_02	R & I	The <u>MS</u> word format allowed for intermediate and draft versions is not applicable for this test plan



	Document- No. <b>AE- PL- KNMI- L2BP- 001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>49/ 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				

	<i>requirement</i>	<i>Test Case Name</i>	<i>Verifi- cation Method</i>	<i>Software Component/remark</i>
	<del>purpose of change tracking.</del>			
[L2BP 10]	<del>The L2BP software will be maintained and warranted against design defects from delivery through to the launch of the satellite plus 39 months deleted</del>	-	D	<del>all modules Warranty and maintenance are defined in [RD6]</del>
[L2BP 20]	A new release of the L2BP software will take less than 4 hours to install	L2bP_GEN_03	D	all modules
[GEN 70]	<del>deleted</del>			
[GEN 80]	The L2BP systems will be built upon a commercial Linux Operating System (OS). Upgrades to the versions of the OS and other COTS and public domain software initiated by ECMWF will be mutually agreed between ECMWF and ESA at least 2 months prior to a planned software delivery.	L2bP_GEN_03	D	all modules  Note that a specific target platform is defined in CN1 sow appendix A1.2 <del>(is this [AD 3]?)</del> . As a minimum the code shall install and run on that system.
[GEN 90]	The L2BP system will be designed to use files and directory structures for storage and management of system data	L2bP_EXT_01	D	all modules
[GEN 100]	The L2BP software will be designed to trap mathematical exceptions	L2bP_GEN_04	R & I	all modules
[GEN 110]	When a mathematical exception has been triggered, a warning log msg will be issued, if execution is allowed to continue	L2bP_GEN_04	T	all modules
[GEN 120]	When a mathematical exception	L2bP_GEN_04	T	all modules



	Document- No. <b>AE- PL- KNMI- L2BP- 001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>50/ 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				

	<i>requirement</i>	<i>Test Case Name</i>	<i>Verifi- cation Method</i>	<i>Software Component/remark</i>
	has been triggered, a warning log msg will be issued, if execution is terminated.			
[GEN 130]	Error and warning msgs will identify the module or function where the exception occurred (when possible).	L2bP_GEN_04	T	all modules
[L2BP 30]	The L2BP shall process to completion each JobOrder for which valid L1B observations are available <sup>2</sup>	L2bP_HMI_01	T	L2BP
[L2BP 80]	The L2BP software shall be delivered in binary and source code formats including all scripts and libraries required for the installation, <u>plus binary data needed for testing.</u>	L2bP_GEN_03	D	L2BP [creating a binary version for delivery is not possible at KNMI at the moment since we do not have a RedHat platform to generate this. Maybe ECMWF or MeteoFrance can do this?]
[L2BP 90]	The L2BP software delivery shall include a tool for viewing the contents of each input Auxiliary Data File	L2bP_TOOLS_01	D	the different TestRead* tools
[L2BP 100]	The L2BP software delivery shall include instructions for constructing a Job Order	L2bP_EXT_02	R & I	-
[L2BP 110]	The L2BP software delivery shall include a tool for viewing the contents of each L2BP product.	L2bP_PROD_04	D	the L2BC_file_handling module
[L2BP 120]	<u>n.a. The L2BP software delivery shall include a tool for graphical display of wind data from a Level 2b product</u>	<del>[deleted]</del> t.b.d.	D	
[L2BP 130]	The L2BP software delivery shall include a tool for graphical display of calibration data from	L2bP_TOOLS_02	D	

2 Criteria for valid observations are specified in the Aeolus Level 2B Processor design document [RD 1]. The criteria include no-process decisions based on invalid or missing data.





	<i>requirement</i>	<i>Test Case Name</i>	<i>Verifi- cation Method</i>	<i>Software Component/remark</i>
	Auxiliary Calibration products			
[L2BP 140]	The L2BP software delivery shall include the input data and output products used in the L2BP factory acceptance test.	L2bP_DEL_01	R & I	-
[L2BP 150]	The L2BP software delivery shall include the installation scripts and tools required to execute the L2BP factory acceptance test and evaluate the test results	L2bP_DEL_01	D	FAT script
[L2BP 160]	One representative set of input Auxiliary Data Files shall be provided with the L2BP system delivery	L2bP_DEL_01	R & I	-
[L2BP161]	The L2bP software shall include functionalities to generate or update any L2B input auxiliary <del>data</del> files	L2bP_TOOLS_03	D	
[L2BP162]	The L2bP software shall provide a functionality to convert L2B products in EE format to ascii (for a user selected ADS, MDS, and range of records)	L2bP_PROD_04	D	WriteL2BtoTxt in directory L2BC_file_handling
[L2BP 170]	The L2BP delivery shall include a user manual in Adobe Portable Document Format (PDF)	L2bP_DEL_01	R & I	-
[L2BP 200]	<del>The L2BP shall be operated by one person.<sup>3</sup>deleted</del>	<del>L2bP_HMI_01</del>	<del>⌘</del>	
[L2BP 210]	<del>The L2BP shall allow continuous operation, i.e. to perform a new processing task after completion of the previous taskdeleted</del>	<del>L2bP_HMI_01</del>	<del>⌘</del>	(note that the SRD needs to be adapted to this new formulation)
[L2BP 220]	<del>The L2BP installation shall allow to protect code, libraries, databases, processing results from deletion/modification by unauthorized users.deleted</del>	<del>[not yet implemented, this probably needs a new testcase]</del>	<del>R &amp; I</del>	(note that the SRD needs to be adapted to this new formulation)

---



3Operation shall also be possible in a stand alone configuration, from command line (see email ESA, by H. Nett, 31- jan- 2007)



	Document- No. <b>AE- PL- KNMI- L2BP- 001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>52/ 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				



	<i>requirement</i>	<i>Test Case Name</i>	<i>Veri- fication Method</i>	<i>Software Component/remark</i>
[L2BP 230]	It shall be possible to install and change configuration settings of the L2bP by a user with no administrator privileges.	L2bP_GEN_03	D	<del>Install and edit some settings as normal user to demonstrate this</del>
[L2BP 250]	The L2BP shall accept as input any valid Level 1b product provided by the Front- End- Processor ( <del>FEP</del> ) of the PDS. The content of the Level 1b product is defined in [RD 2].	L2bP_HMI_01	D	L2BP
[L2BP 260]	The L2BP shall produce a Level 2B formatted product as specified in the Aeolus L2B/L2C Processor IODD [AD 8].	L2bP_PROD_04	T	the L2BC_file_handling module ( <del>see also [L2BP-700]</del> )
[L2BP 270]	The L2BP shall accept an Auxiliary L2B Processing Parameters file as specified in the Aeolus L2B/L2C Processor IODD [AD 8]	L2bP_HMI_01	T	L2BP
[L2BP 280]	The L2BP shall accept an Auxiliary Climatology file as specified in the Aeolus L2B/L2C Processor IODD [AD 8]	L2bP_HMI_01	T	L2BP
[L2BP 290]	The L2BP shall accept an Auxiliary Rayleigh- Brillouin Calibration file as specified in the Aeolus L2B/L2C Processor IODD [AD 8]	L2bP_HMI_01	T	L2BP
[L2BP 300]	the L2BP shall accept <u>an Auxiliary L2b Meteorological Parameters auxiliary files as specified in the Aeolus L2B/L2C Processor IODD [AD 8], irrespective of whether the file is created for the "nominal" case (Mmeteorological profile s-geolocations aligned with observation geo-locations from in the input L1B input product) and or for the "late processing" case (meteorological profile geo-locations created of Met profiles created from orbit prediction data) as specified in the Aeolus L2B/L2C Processor IODD [AD 8].</u>	L2bP_HMI_01	T	L2BP



	Document- No. <b>AE-PL-KNMI- L2BP- 001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>53/ 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				

	<i>requirement</i>	<i>Test Case Name</i>	<i>Veri- fication Method</i>	<i>Software Component/remark</i>
[L2BP 301]	The L2BP shall accept an Auxiliary Calibration Parameters file as specified in the L2A IODD [RD 3]	L2bP_HMI_01	T	L2BP (see section 4.3 in the L2A IODD, [RD 3]. Note that this file still needs to be defined)
[L2BP 302]	The L2bP shall be capable of processing a L1B input product (EE or BUFR format) containing up to 8 orbits worth of data.	L2bP_PROD_03	T	L2BP
[L2BP 310]	The L2BP shall provide for processing a Job Order initiated by the Thin Layer	L2bP_HMI_01	T	L2BP
[L2BP 320]	The L2BP Job Order files- <del>accepted by the L2BP</del> shall be formatted as specified in the Aeolus L2BP External ICD [AD 9]	L2bP_EXT_02	R & I	-
[L2BP 330]	The L2BP shall produce an Exit Code upon the completion or termination of each processing job as defined in the Aeolus Level 2B External ICD. [AD 9]	L2bP_HMI_01	T	L2BP
[L2BP 335]	The L2BP shall produce a Product List file for each L2BP processing Job Order as defined in the Aeolus L2B External ICD [AD 9]	L2bP_EXT_01	T	L2BP
[L2BP 340]	The L2BP shall provide L2BP Informational messages on the stdout <del>and</del> /or stderr stream	L2bP_EXT_03	T	L2BP (not in use yet)
[L2BP 350]	The L2BP shall provide L2BP Progress messages on the stdout <del>and</del> /or stderr stream	L2bP_EXT_03	T	L2BP
[L2BP 360]	The L2BP shall provide L2BP Warning messages on the stdout <del>and</del> /or stderr stream	L2bP_EXT_03	T	L2BP
[L2BP 370]	The L2BP shall provide L2BP Error messages on the stdout <del>and</del> /or stderr stream	L2bP_EXT_03	T	L2BP (only if an error is provoked)
[L2BP 380]	The L2BP shall provide L2BP Debug messages on the stdout <del>and</del> /or stderr stream	L2bP_EXT_03	T	L2BP (not in use yet)
[L2BP 381]	It shall be possible to write log/warning/error messages to a log file (for use when the L2bP is not operated via the ThinLayer)	L2bP_HMI_01	T	



	Document- No. <b>AE- PL- KNMI- L2BP- 001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>54 / 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				

	<i>requirement</i>	<i>Test Case Name</i>	<i>Verifi- cation Method</i>	<i>Software Component/remark</i>
[L2BP 382]	It shall be possible to generate a 'short' log file (messages for low-level processing steps suppressed)	L2BP_HMI_01	D	
[L2BP 383]	It shall be possible to view log/warning/error data during execution of a processing run	L2bP_EXT_01	D	
[L2BP 390]	One Workstation Configuration file shall be provided with the L2BP system delivery as specified in the Aeolus Level 2BP External ICD [AD 9]	L2bP_EXT_02	R & I	
[L2BP 400]	The L2BP Workstation Configuration file shall list each L2BP processing service	L2bP_EXT_02	R & I	
[L2BP 410]	At least one Task Table shall be provided with the L2BP system delivery as specified in the Aeolus Level 2B Processor External ICD [AD 9]	L2bP_EXT_02	R & I	
[L2BP 420]	The L2BP Task Table file shall define the L2BP processing tasks required to provide a processing service	L2bP_EXT_02	R & I	
[L2BP 430]	The L2BP Task Table file shall define the input and outputs for each L2BP processing task.	L2bP_EXT_02	R & I	
[L2BP 440]	Each L2B processing task shall be initiated as a command line task with the Job Order filename as the only parameter.	L2bP_HMI_01	D	L2BP
[L2BP 450]	The L2BP shall provide for stopping each processing task by a kill or terminate signal. Under normal conditions, the different tasks constituting a processor shall terminate on their own without the need to receive a kill or terminate signal.	L2bP_EXT_03	D	L2BP
[L2BP 460]	The L2BP shall produce a Product Report file upon completion of each processing Job Order, as specified in the Aeolus L2B External ICD [AD 9]	L2bP_EXT_01	D	
[L2BP 470]	The L2BP shall provide for processing a Job Order initiated by the Thin Layer Emulator	L2bP_EXT_01	D	L2BP



<i>requirement</i>		<i>Test Case Name</i>	<i>Verifi- cation Method</i>	<i>Software Component/remark</i>
[L2BP 480]	The L2BP shall use the Thin Layer Emulator <u>as a Thin Layer emulation interface</u> for testing the L2BP independently of the rest of the PDS	L2bP_EXT_01	D	L2BP
[L2BP 490]	Ways to operate the L2BP through interfaces different from the Thin Layer shall be specified in the L2BP Software User Manual [RD 5]	L2bP_GEN_02	R & I	-
[L2BP 500]	The L2BP shall process all complete observations in a Level 1b product	L2bP_HMI_01	D	L2BP <sup>4</sup>
[L2BP 510]	The L2BP shall process all complete observations in a Level 1b product that are within the start and end times specified in the Job Order	L2bP_HMI_01	D	<del>L2BP (at the moment we do not yet take the times given in the JobOrder file into account, and just process everything in the L1B file)</del> TODO: add this in the post-FAT clean-up version
[L2BP 520]	The L2BP shall process valid observations in a Level 1b file to produce an Aeolus L2B product	L2bP_HMI_01	D	L2BP (see remark [L2BP 500])
[L2BP 530]	The L2BP will use the Level 2B processing algorithms specified in the L2B ATBD [RD 4].	L2bP_GEN_02	R & I	all modules
[L2BP 540]	The L2BP shall detect missing or corrupted data in the input Level 1b product and continue to process remaining valid data <sup>5</sup>	L2bP_PROD_02	D	L2BP
[L2BP 550]	The L2BP shall stop processing and produce an error message if any input file is not accessible prior to the start of an L2BP processing task	L2bP_PROD_02	D	L2BP
[L2BP 560]	The L2BP shall log messages in the L2BP event log listing each input file that was not accessible to a started L2BP executable	L2bP_PROD_02	D	L2BP (At the moment we only detect and report the first problematic file) TODO: add reporting of ALL



4 Note that if a L1B product contains a partial BRC, and the file is not corrupt (which is a different testcase) then this should be reflected in the PCD. therefore the L2BP inspects the PCD and processes all valid L1B data in the input file.

5 As stated in the footnote with requirement [L2BP 30] criteria for valid L1B observations are specified elsewhere. Nonetheless, the L2BP shall exhibit the behaviour defined in Table 4.22.



<i>requirement</i>		<i>Test Case Name</i>	<i>Verifi- cation Method</i>	<i>Software Component/remark</i>
				<del>missing input files in the post-FAT cleanup-version(the L2BP uses the stdout/stderr channel for logging, as specified by the ThinLayer interface-definition).</del>
[L2BP 570]	If an L2BP processing run fails irrecoverably, the L2BP event log will list the last successfully completed processing step and no further events will be listed for that JobOrder	L2bP_PROD_02	D	L2BP (of course the error condition will be reported as well)
[L2BP 580]	The L2BP Product Report shall contain the version number of the L2BP executable software	L2bP_PROD_01	R & I	L2BP
[L2BP 590]	The L2BP Product Report shall contain the names of the input files.	L2bP_PROD_01	R & I	L2BP (not implemented yet in logging format, only some ascii test prints) TODO: add this in the post-FAT clean-up version
[L2BP 600]	The L2BP Product Report shall contain the names of the generated product files.	L2bP_PROD_01	R & I	L2BP (not implemented yet) TODO: add this in the post-FAT clean-up version
[L2BP 610]	The L2BP Product Report shall contain the size of each generated product file.	L2bP_PROD_01	R & I	L2BP (not implemented yet) TODO: add this in the post-FAT clean-up version
[L2BP 620]	The L2BP Product Report shall contain an exit code indicating success or failure	L2bP_PROD_01	R & I	L2BP (not implemented yet) TODO: add this in the post-FAT clean-up version
[L2BP 630]	The L2BP Product Report shall contain a list of errors and warnings if applicable	L2bP_PROD_01	R & I	L2BP
[L2BP 640]	The L2BP Product Report shall include the start and stop time of operation	L2bP_PROD_01	R & I	L2BP (datetime-stamps are part of the logging format) TODO: add dedicated log messages for start and stop in the post-FAT clean-up version
[L2BP 650]	The L2BP shall generate log messages indicating L2BP processing task status	L2bP_PROD_01	D	L2BP (not implemented yet, what status should we report on? Is this the same as the progress messages we can write to the log?)
[L2BP 660]	The L2BP shall generate log	L2bP_PROD_02	D	L2BP



	Document- No. <b>AE- PL- KNMI- L2BP- 001</b>	Issue: <b>V 1.32</b>	Date: <b>11.04.2007</b>	Page: <b>57/ 62</b>	
	Doc.- Title: <b>Aeolus Level 2B Processor Software Verification &amp; Validation Plan</b>				

	<i>requirement</i>	<i>Test Case Name</i>	<i>Verifi- cation Method</i>	<i>Software Component/remark</i>
	messages indicating L2BP process warnings and errors			
[L2BP 670]	The L2BP shall provide the event log data stream through the stdout and stderr	L2bP_EXT_03	D	L2BP
[L2BP 680]	The L2BP software shall generate log messages for the Informational/Warning/Error events specified in the Aeolus L2BP external ICD [AD 9].	L2bP_PROD_01	D	L2BP (we need to add this list to the ICD)
[L2BP 690]	The L2BP software shall store event log data in a log file per Job Order	<u>L2bP_HMI_01</u> <del>[not yet implemented]</del>		The ThinLayer is responsible for storing the stdout stream in the log file. <del>Or should we produce a private log file as well, holding the same data?</del> Yes, according to ESA TODO: add this in the post-FAT clean-up version
[L2BP 700]	<del>The L2BP shall generate a L2B product as specified in the Aeolus L2BP IODD [AD 8]. duplicated requirement deleted</del>	L2bP_PROD_04	<del>T</del>	see also [L2BP 260]
[L2BP 710]	Each L2BP product shall be annotated with the Job Order Type as Nominal Processing, Reprocessing or Backlog Processing	L2bP_PROD_04	T	L2BP (probably defined in the filename at the moment) (not yet implemented, this means we have to add an element to the L2BC SPH definition) TODO: check how this is done in the L1B/L2A (see email H. Nett, 31- 1- 2007)
[L2BP 720]	Each L2BP product shall annotate quality statistics as specified in the Aeolus L2B/L2C processor IODD [AD 8].	L2bP_PROD_04	D	L2BP (note: most of the L2B SPH items are not yet filled) TODO: fill these fields in the post-FAT clean-up version
[L2BP 730]	Each L2BP product shall contain a Main Product Header (MPH) as specified in the Aeolus L2B/L2C processor IODD [AD 8]	L2bP_PROD_04	D	L2BP
[L2BP 740]	Each L2BP product shall contain a Specific Product Header (SPH) as specified in the Aeolus L2B/L2C processor IODD [AD 8]	L2bP_PROD_04	D	L2BP





<i>requirement</i>		<i>Test Case Name</i>	<i>Verifi- cation Method</i>	<i>Software Component/remark</i>
[L2BP 760]	Each L2BP product shall contain at least one Data Set Descriptor as specified in the Aeolus L2B/L2C processor IODD [AD 8]	L2bP_PROD_04	D	L2BP
[L2BP 770]	The L2bP shall be capable of processing a full orbit worth of L1B data (EE or BUFR format) within 30 min. (tbd) on the specified <b>reference</b> target platform.	L2bP_PERF_01	T	L2BP
[L2BP 780]	The L2BP <u>code and standalone applications</u> shall be <u>installable and</u> run on a <u>single processor</u> Pentium 4-PC (clock rate 2.5GHz) with a minimum of 256 Mbytes <u>of free RAM space</u> and <u>a useble diskspcace of 20.500 Mbytes of free disk space, and the following configuration:</u>  <u>OS: RedHat Linux Enterprise 3 (update 2) Server, kernel 2.4.21</u>  <u>Compiler: GNU Compiler Collection (gcc) v.3.3.2</u>  <u>Python: version at least 2.2</u>	L2bP_GEN_01	D	check the hardware
[L2BP 790]	The L2BP shall be programmed in Fortran <u>90</u> and/or Ansi C/C++ and/or Python, and or <b>csh/sh scripts and Makefiles.</b>	L2bP_GEN_02	R & I	<b>inspect the source code</b>  <b>(note that we need to update the SRD for this item)</b>
[L2BP 800]	A list of any non- standard functions/modules (compiler- specific) shall be maintained and included in the <b>D</b> esign <b>D</b> ocument	L2bP_GEN_02	R & I	<b>)all our non- standard code is collected in a single module called compiler_features located in the support directory.</b>
[L2BP 810]	The <u>L2BPauxiliary tools for viewing L2BP products and associated auxiliary data</u> shall make use of the following <b>commercial tools (TBC):</b> <b>Matlab</b> . <u>(might be replaced by the free SCILAB, Octave or Rlab, but these are not 100% compatible, so some effort is needed to do this)</u> <u>This commercial tool shall not be used for:</u>  <u>- the L2BP inself</u>  <u>- the L1B EE2BUFR convertor</u>	L2bP_GEN_02	R & I	<b>Matlab is only needed if we are required to give graphical output. If text output is sufficient we have all the fortran tools available.</b>



<i>requirement</i>		<i>Test Case Name</i>	<i>Verifi- cation Method</i>	<i>Software Component/remark</i>
??[L1B- EE2BUFR 10]	<p>requirements on the EE2BUFR tool which converts L1B files from EE to BUFR format.</p> <p>This should probably be split into the following items:</p> <ul style="list-style-type: none"> <li>-tool compilation and installation</li> <li>-tool execution</li> <li>-inspection of the outputs, and objective verification of the numerical content of the redirected output, by using the supplied diff tool</li> <li>-execution of the L2bP on the generated BUFR file to demonstrate correct production of an L2B product</li> </ul> <p>The L1B EE2BUFR software shall be delivered as part of the Level 2B Processor software delivery</p>	L2bP_TOOLS_04	VF	Note that no requirements have been specified in the current version of the SRD so I postpone the precise definition of this testing until these requirements are available in a next update of the SRD
[L1B- EE2BUFR 20]	It shall be possible to install and test the L1B EE2BUFR software as a subset of the L2B installation	L2bP_TOOLS_04 -b.d-	I	L2B_ee2bufr
[L1B- EE2BUFR 30]	The L1B EE2BUFR software shall provide a functionality to convert L1B BUFR-format files to Ascii (for user selected ADS, MDS and range of records).	L2bP_TOOLS_04 -b.d-	I	L2B_ee2bufr
[L1B- EE2BUFR 80]	The L1B EE2BUFR convertor shall accept as input any valid Level 1b product provided by the LTA. The content of the Level 1b product is defined in [RD 2]	L2bP_TOOLS_04 -b.d-	I	L2B_ee2bufr
[L1B- EE2BUFR 90]	The L1B EE2BUFR convertor shall produce a Level 1B BUFR-product formatted as specified in [AD 10]	L2bP_TOOLS_04 -b.d-	I	L2B_ee2bufr
[L1B- EE2BUFR 110]	The L1B EE2BUFR convertor shall be operated from a Unix shell using a command-line instruction as specified in the L1B EE2BUFR User Instructions [AD 11]	L2bP_TOOLS_04 -b.d-	I	L2B_ee2bufr
[L1B- EE2BUFR 120]	The L1B EE2BUFR convertor shall produce an Exit Code upon the completion or termination of each processing	L2bP_TOOLS_04 -b.d-		not yet implemented





requirement	Test Case Name	Verifi- cation Method	Software Component/remark
	job as defined in the L1B EE2BUFR User Instructions [AD 11]		
[L1B- EE2BUFR 170]	The L1B EE2BUFR convertor shall process all complete observations in a Level 1b product	L2bP_TOOLS_04+ T .b.d-	L2B_ee2bufr
[L1B- EE2BUFR 190]	The L1B EE2BUFR convertor shall detect Level 1b data violating the L1B IODD [RD 2] and continue to process remaining valid data	L2bP_TOOLS_04+ T .b.d-	L2B_ee2bufr

**Table 4.21 : Tracability or Cross- Reference matrix**

Invalidity Case	L2BP Behaviour	Comments
No valid observations in L1B input (or in the processing window specified in the jobOrder)	L2B outputs produced with consistent header data and dataset structure	
Partially valid data (Mie or Rayleigh data unavailable at some altitudes)	L2B outputs produced whenever processing algorithms (as selected by AUX_PAR_2B processing parameter settings) permit	
Failed matchup of L1B input data and AuxMetData	L2B outputs produced with consistent header data and dataset structure	TBC
Misalignment of Mie and Rayleigh range bins	Regarded as aligned provided misalignment is within tolerances	

Table 4.22.: L2BP behaviour on invalid L1B data



	Document- No.	Issue:	Date:	Page:	
	AE- PL- KNMI- L2BP- 001	V 1.32	11.04.2007	61 / 62	
	Doc.- Title:				
	Aeolus Level 2B Processor Software Verification & Validation Plan				

## B Test Tools

The following sections describe the additional tools that are used in the tests described in this document.

### B.1 unix tools

A number of common unix/linux tools are used. These will be installed on almost every linux/unix system, but still it should be checked whether they are installed before starting the FAT. Free versions of these tools can be found on the internet in many places.

- a text editor, any editor will do, for example: vi, (x)emacs, nedit, zile, pine
- a difference generator, for example: diff, xdiff or xxdiff (xxdiff is recommended)
- a tool to view binary files, for example: hexdump
- a tool to edit binary files, for example: hexedit (needed to test the robustness of the file reading code in case of input files that violate the IODD)
- more?

### B.2 external tools

The following external tool isis is used in the tests described in this document.



- The ThinLayer emulator

### B.3 custom tools

The following custom tools are provided with the L2BP software delivery.

- TestReadAMDdata: read an AMD datafile in EEformat to verify it
- TestWriteAMDdata: write an AMD datafile in EEformat, using ascii input
- TestWriteAuxClimData: write an AuxClim datafile in EEformat, using a single predefined profile
- L1B\_ee2bufr: convert an L1B product file from EEto BUFRformat
- TestL1B\_ee2bufr: read an L1B product file in BUFRformat to verify it
- WriteSomeNumbers: write a binary file in native byteorder format
- TestReadL1Bdata: read an L1B product file in EEformat to verify it
- Test\_Read\_L2B\_AuxPar\_file: read an L2B\_AuxPar file in EEformat to verify it
- TestWriteL2BCdata: write a L2B/C datafile holding dummy data in EEformat
- TestReadL2BCdata: read a L2B/C datafile in EEformat to verify it
- Write\_L2BC\_Text\_Product: convert a L2B/C file to ascii
- GenerateRBCdata: generate an Aux\_RBC file in EEformatting
- TestWriteRBCdata: write an Aux\_RBC datafile holding dummy data in EEformat
- TestReadRBCdata: read an Aux\_RBC file in EEformat to verify it
- run\_L2BP.sc: filter stdout output of the processor, and let only properly formatted log messages pass



	Document- No.	Issue:	Date:	Page:	
	AE- PL- KNMI- L2BP- 001	V 1.32	11.04.2007	62 / 62	
	Doc.- Title:				
	Aeolus Level 2B Processor Software Verification & Validation Plan				

- TestReadJobOrderData: read a JobOrder file and verify it
- difftool: compare 2 files, taking numerical differences between real numbers into account, upto a user specified accuracy.

## B.4 matlab tools

The following matlab tools are provided with the L2BP software delivery.

- CreateAuxMet.m: import AMD data from E2S output files
- NoisePhotoCountL1A.m: add noise to L1A data
- RBCorrection.m: Apply the Ryaleigh- Brillouin correction scheme to L1B data
- ReadAuxMet.m: import an AMD file in EE format
- ReadKNMIData.m: read NWP data from KNMI ascii format
- ReadL1AData.m: import an L1A product file in EE format
- ReadL1BData.m: import an L1B product file in EE format
- ReadL2BAMD.m: import an AMD file in EE format
- ReadL2B.m: import an L2B product file in EE format
- ReadL2BRBC.m: import an RBC file in EE format
- WriteKNMIData.m: write NWP data in KNMI ascii format
- WriteL1AData.m: write an L1A product file in EE format